Computational methods to explore hierarchical and modular structure of biological networks

by

Yongjin Park

A dissertation submitted to The Johns Hopkins University
in conformity with the requirements for the degree of
Doctor of Philosophy

Baltimore, Maryland
February, 2014

Networks have been widely used to understand structure of complex systems. From studying biological networks of protein-protein, genetic and other types of interactions, we gain insights into functional organization of static biological systems that could hardly be measured experimentally in current state-of-the-art technology. Biological networks also serve as a principled framework that integrates multiple sources of genome-wide data sets such as gene expression arrays and sequencing. Yet, a large-scale network is often intractable for intuitive visualization and computation.

We developed novel network clustering algorithms to harness the power of genome-scale biological networks of all genes/proteins. Especially our algorithms were capable of finding hidden modular structures in hierarchical stochastic block model. Since the modules are organized hierarchically, our algorithms facilitate downstream analysis and design of in-depth validation experiments in "divide-and-conquer" strategy. Moreover, we present empirical evidence that the hierarchical and modular structure best explains observed biological networks.

We used the static clustering methods in two ways. First we sought to extend the static methods to dynamic clustering problems, and observed general patterns of dynamics of network modules. For examples we demonstrate dynamics of yeast metabolic cycle and Arabidopsis root developmental process. Moreover, we propose a prioritization scheme that sorts identified network modules in the order of discriminative power.

In the course of research we conclude that biological networks are best understood as hierarchically organized modules, and the modules remain stable in unperturbed biological process, but they can respond differently to abnormal / external perturbations such as knock-down of key enzymes.

This dissertation is dedicated to my father, YoungSik Park, who willingly abandoned his opportunity to Ph.D. for his beloved family. Truly every moment was blessing to me.

My years at the Johns Hopkins University could not be made without the help of wonderful colleagues, friends and family. First of all, I would like to thank my advisor, Professor Joel S. Bader, for his insightful guidance and allowance of intellectual freedom. Throughout the course of my graduate research I have stumbled upon many obstacles, due to my mistakes or lack of experience, but Professor Bader has always been supportive and patient and that turned mistakes to wonderful research experience. Moreover, I want to thank my thesis committee, Professors Aleksander S. Popel and Mark H. Dredze. Writing a thesis was not an easy task for me but their guidance helped organize structures and made ideas more deliverable. I hope collaboration with my mentors may continue throughout my research career. I also want to thank kind and smart people in the Bader laboratory. Especially I had a great time with Drs. Giovanni Stracquadanio and Elisa Pappalardo from Italy. They have brought my family home and served best of Italian culture. Last but not least, I was fortunate enough to start a family during my graduate years. I married to a wonderful woman, Dr. Hee Yeon Im, and we have most adorable daughter, Elise Park. Family always reminds me the very best reason for doing science. Without family I would have not been able to finish this work.

Research is my honest way of expressing homage to the Creator, who established fundamentals of universe and mechanisms leading to intricate complexity and diversity of life.

> *O Lord, our Lord, how majestic is your name in all the earth! You have set your glory above the heavens* (Psalm 8:1, Bible, English Standard Version).

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Biological systems are complex, but modular [66]. Genes, proteins and other molecules interact with each other, and respond to many environmental queues, and produce wide spectra of phenotypes. Networks represent these interactions mathematically and graphically. Of many benefits we gain from studying networks we want to focus on modularity of the system, from which observed networks were generated. We aim to uncover functional modules of biological entities, especially genes/proteins, just as we can infer human organizations from social networks.

The notion of a functional module may seem subtle, but can be defined as a set of genes/proteins of which functions are better understood as a set, not individually [66]. Biological networks have long been suggested as a primary tool to identify the functional modules [157, 173]. In general network analysis, modules are defined as a densely connected subgraph, compared to background [108], therefore members in the same module form "a small world" [193]. In social network studies, modules are termed groups or communities. In classical studies, communities are directly related to factions of people [202] and groups of coworkers [130, 131]. Similarly densely connected subgraphs found in biologi-

cal networks were considered functional modules, and it was demonstrated that genes in the same modules share similar functional annotations [8,9,35,169,198].

1.1 Biological networks

Physical interactions The Yeast Two Hybrid assay (Y2H) indirectly measures binding of two proteins in a genetic system [43]. The native GAL4 protein consists of two domains, the N-terminal domain binding to DNA and the C-terminal domain that activates downstream transcription of a reporter gene. Idea is to genetically engineer the GAL4 and create a hybrid protein X bearing with the GAL4's N-terminal domain and the other protein with the C-terminal domain, so that we may observe expression of the reporter gene only if two proteins, X and Y, bind to make the N- and C-terminal domains work functionally. This technique was successfully applied to identify genome-scale interaction networks of multiple species, *Saccharomyces cerevisiae* [80,187], *Drosophila melanogaster* [54] and *Homo sapiens* [161].

The Tandem Affinity Purification (TAP) method identifies a protein complex, rather than a pair of binding proteins. The TAP method begins with engineering a target protein, fused with affinity purification tag, and specifically selects out protein complex attached to the target [156]. Followed by mass spectrometry and database search, we can quickly identify attached proteins [51].

A similar high-throughput biochemical assay works for identifying DNA sequences bound to a specific target protein, which can be isolated by chromatin immunoprecipitation (ChIP) [53]. From ChIP followed by high-throughput microarrays (ChIP-chip) [21] or short-read sequencing (ChIP-seq) [15, 83, 122], we

obtain genome-wide protein-DNA binding profiles, yielding a protein-DNA network.

These techniques are complementary to each other. For instance, the TAP-MS is clearly more advantageous in screening *in vivo* interactions that occur naturally in live cells [98]. However, the Y2H tends to produce more accurate interaction maps, especially in yeast cells [201].

Genetic interactions Genetic interactions, broadly termed epistasis, disclose gene-gene interactions, which can be seen only through genetic perturbations, but hidden in the wild type [149]. A degree of the genetic interaction between genes x and y is usually quantified by difference between observed fitness score F_{xy} of the double mutant cell and expected fitness $\mathbb{E}\left[F_{xy}\right]$. A significant change made by double mutant, i.e., $F_{xy} \ll \mathbb{E}\left[F_{xy}\right]$ or $F_{xy} \gg \mathbb{E}\left[F_{xy}\right]$, yields a negative or positive genetic interaction. The test of significance may have to adjust to different definitions of the expectation $\mathbb{E}\left[F_{xy}\right]$ [117] and number of tests as well.

The Synthetic Genetic Array (SGA) [182, 183] was the first example of large-scale double mutant screening. Edges between genes are determined by a certain threshold, or p-value. Alternatively, we may just use quantitative pairwise scores treating them a full real-valued matrix, rather a sparse $0/1$ matrix [98, 162, 166]. The technology is scalable enough to scan entire pairs of genes in yeast cells [31]; a similar technique was also applied to mammalian cells [158]. Moreover, we may introduce differential conditions to the cells to resolve condition-specific genetic interactions [12, 63].

Inferred interactions We may as well estimate connectivity by statistical inference. For instance, from a large compendium of gene expression matrices we can calculate correlation coefficient between genes, encompassing may different experiments and conditions [126]. Moreover, pairs of genes participating in known pathways/reactions have a high chance of interaction physically and genetically [32].

1.2 Notations

We will use terms "network" and "graph" interchangeably, since we mainly discuss undirected and unweighted network data, which is equivalent to a graph.

Definition 1 (network). *A graph $G = (V, E)$ is a tuple of sets V and E. The vertex set V takes genes/proteins as an element; the edge set takes pairs of vertices as an element, and $E \subseteq V \times V$. Since edges are undirected, we have $(u, v) \equiv (v, u)$ for any $(u, v) \in E$.*

Equivalently, we define a symmetric $n \times n$ adjacency matrix $A \in \{0, 1\}^{n \times n}$ to represent an undirected and unweighted network.

Definition 2 (adjacency matrix). *A network $G = (V, E)$ has an equivalent $n \times n$ matrix representation such that an element A_{ij} takes 1 if and only if a pair i and j are connected, i.e., $(u, v) \in E$, but takes 0 otherwise.*

We will use n and m repeatedly to denote number of vertices and edges of a given network.

Definition 3 (number of vertices and edges). *Given a network $G = (V, E)$, n, or $n(G)$, corresponds to the number of elements in the vertex set V; m, or $m(G)$, corresponds to the number of elements in the edge set E. Compactly, $n(G) = |V|$ and $m(G) = |E|$.*

We will also use d_i to denote a vertex degree of a vertex i, and D a diagonal matrix of degree sequence.

Definition 4 (vertex degree)**.** *Given a network $G = (V, E)$, d_i denotes the degree of a vertex i, that is a number of neighbors interacting with i. More precisely,*

$$d_i = |\{(a, b) \in E : \ a = i \vee b = i\}|,$$

or

$$d_i = \sum_{j \neq i} A_{ij}.$$

Let D denote a diagonal matrix of degree sequence. More precisely, an element

$$D_{ij} = \begin{cases} d_i & \text{if } i = j, \\ 0 & \text{if } i \neq j. \end{cases}$$

1.3 General problems of network

Here we briefly review problems of network that we may put into the following categories:

- *Test of randomness.*
- *Graph cut / clustering.*
- *Graph search / prioritization.*

Tests of randomness

A firsthand analysis of a real-world network usually begins with measurement of statistics, or quantification of repeated patterns. Examples of statistics include

a sequence of vertex degrees, clustering coefficients [76, 193] and betweenness centrality [47]. Examples of patterns include triangles, feedback and feedfoward loops [123], stars [13] and cliques [8, 136]. However, these statistics / patterns may occur by chance and need to be thoroughly tested in a formal framework such as statistical hypothesis test.

In order to construct a hypothesis test, a notion of randomness, or the null distribution, has to be set up beforehand. Random graph models well serve for this purpose. Examples of random graph models include the Erdös and Rényi's model [41, 42], the small world network [193] and the preferential attachment model [14]. Given a class of patterns and the null model, now we can test whether the empirical statistics unusually diverge at a certain level of p-value [123]. However, one should note that different models emphasize different aspects of a network [107], and statistical tests only reject the null hypothesis, not the other way.

Graph cut / clustering

Classical graph cut problems A graph cut problem has rich history and has been approached from diverse disciplines, pure mathematics, combinatorial optimization and computer science. The goal is to find an optimal partition of vertices, that divides V into disjoint subsets S and $\bar{S}$, such that $S \cup \bar{S} = V$ and $S \cap \bar{S} = \emptyset$. Edges between S and $\bar{S}$ are called cut-edges, i.e., $E_C \equiv \{(u, v) \in E : u \in S \wedge v \in \bar{S}\}$. There is a weight function maps edges to some value, i.e., $f : (u, v) \to \mathbb{R}$. In undirected and unweighted network, $f(u, v) = f(v, u) = 1$ for any $u \neq v$, but $f(u, u)$ is unnecessary. We may choose $S \subset V$ either to maximize or minimize total weight of the resulting cut-edges, i.e., $W \equiv \sum_{(u,v)\in E_C} f(u, v)$.

Although both directions appear in the same framework, the minimization problem has a polynomial time algorithm while the maximization is provably NP-complete [57]. Due to dual relationship with the maximum flow problem [44], we may resolve the min-cut using an algorithm that solves the dual flow problem. For instance, a classical augmenting path algorithm solves the maximum flow problem in $O(m^2 \log C)$ time complexity [38], with a constant factor C; a notable randomization algorithm solves in $O(n^2 \log^3 n)$ time with high probability [87]. For the maximum cut problem, we may first solve a relaxed problem, where fractional assignment is allowed, for instance semi-definite programming, then round fractional solutions to find an optimal point [56, 57].

Scalable heuristics based on graph transformation For a large-scale network, fast heuristic methods must be considered. Otherwise, we would not have a solution in practice, or even if we had so, a solution found by exact algorithms could have been misguided by inevitable noise of data collection process. Multiple runs of a fast heuristic method, backed up by bootstrap [88], are generally more applicable than an expensive exact method. Many scalable approaches transform original intractable input data to workable structures. For instance, a large vertex-set V is contracted to V' where $|V'| \ll |V|$; or, a dense adjacency matrix A is "sparsified" so as to increase the number of zero entities. A multi-scaling algorithm exploits the first idea [90], iteratively contracts pairs of vertices. With a high probability an optimal cut found in the contracted network $G' = (V', E')$ is similar to true optimal solution found in $G = (V, E)$ [87], and subsequent fine tuning steps fix mis-classified pairs of vertices in fast local moves. The Markov Cluster Process (MCL) algorithm is an example of the second idea. MCL alternates

two operations, matrix multiplication (inflation) and normalization (expansion), to simplify entangled topology to tree-like components [188], where finding an optimal cut is rather trivial.

The idea of transforming network, or adjacency matrix, has been proposed in classical social network studies, but scalability was largely ignored, and the proposed algorithm treats data as a dense matrix, rather than a sparse matrix [25, 194].

Spectral clustering Spectral clustering was initially attempted to solve a general clustering problem, i.e., clustering of real-valued vectors [116, 134]. A general algorithm begins with computing similarity (or dissimilarity) matrix, then projects the data onto a separable manifold, and there identifies clusters [114]. A graph cut problem directly translates to the spectral clustering by simply replacing similarity matrix with adjacency matrix A. Eigen decomposition of the graph Laplacian,

$$L = D^{-1/2} (D - A) D^{-1/2},$$

computed from adjacency matrix A (Defn. 2) and degree matrix D (Defn. 4), is soft relaxation of the graph cut problem [114]. For scalability we may sparsify similarity matrix or apply fast projection methods [46].

Modularity maximization Instead of finding a minimum cut between clusters, we may enrich edges within clusters. Newman's modularity is such a metric [132]. A basic idea is to use edges discounted by expectation and enrich them

inside. We seek to maximize

$$\text{modularity} \equiv \sum_{u<v} \mathbb{1}[u, v \text{ are in the same cluster}] \, (A_{uv} - \mathbb{E}[A_{uv}]),$$

where $\mathbb{E}[A_{uv}] = d_u d_v / 2m$. There are several scalable algorithms that solve the problem quickly, but approximately [7, 19, 30]. However, the problem is fundamentally NP-complete [24], and bears the resolution limit problem, that separate clusters, i.e., unconnected or very weakly tied components, can easily clump together even in an optimal configuration [45, 59].

Graph search / prioritization

Classical search algorithms, such as the breadth-first-search and the depth-first-search, traverse and rank vertices by a visiting order; random walks on a graph also rank vertices and edges stochastically. In bioinformatic and systems biology researches prioritizing schemes of edges and vertices are indispensable, and most widely used strategy.

Statistical hypothesis testing plays an important role. For instance, we may use p-values, or log-transformed p-value, of hypergeometric test based on number of shared neighborhood between two vertices [58]. Vertex betweenness centrality [1, 47] and edge betweenness centrality [55] were also widely used for social and biological network analysis to identify actors (vertices) and relations (edges) that play an important role in a network. A different flavor of scoring function is graph diffusion kernel, which takes into accounts of global information flow [97, 151].

In systems biology studies we may incorporate contextual information with a raw network structure. The contextual information can be measured from high-

throughput assays of gene activity, or borrowed from prior knowledge base. For instance we may find an optimal subnetwork that differentiates biological conditions of gene expression arrays [36], and may also distinguish types of high-degree vertices based on Pearson's correlation of gene expression vectors [64]; and flow-based ranking may take advantage of other sources of information such as gene ontology [128].

Probabilistic models of networks

Stochastic block models. Stochastic block models encompass all the graph problems we discuses. Stochastic block models combine the idea of graph cut / clustering within a statistical modeling framework [75]; the models fitted to real-world networks may be used to rank importance of nodes and edges.

Breiger *and coworkers* first introduced the notion of "block" in the seminal paper [25]. A block denotes a set of actors, or vertices, and is equivalent to "a cluster" and "a group" as a set. In a matrix context, a block also refers to a (symmetric) sub-matrix of full adjacency matrix.

A motivation behind the blocked structure is justified by the structural equivalence, and that explains why the model can represent an overall network without loss of information. Within a block, vertices are structurally equivalent, and therefore share the same pattern of connectivity [25, 194]. An original algorithm of the block model fitting focuses on finding "zero-blocks" located in between two blocks [25]. These zero-blocks work as locally defined min-cut with weight 0, since there is no edge observed.

Definition 5 (structural equivalence [25, 192, 194])**.** *Two vertices i and j are struc-*

turally equivalent if i interacts with every other vertices in a network G exactly the same as j.

However, in a real-world network where observation is noisy, the structural equivalence may not hold in a strict sense. The stochastic equivalence relaxes the original equivalence to account for stochasticity of data. A property that retains two vertices in a block is now the invariant distribution of edges, rather than the observed edges. Since the model is stochastic, it permits application of general principles of statistics, such as goodness-of-fit tests [192], maximum likelihood and posterior estimation [135, 171].

Definition 6 (stochastic equivalence [75, 192])**.** *Two vertices i and j are stochastically equivalent if and only if the probability of any event on G is unchanged by swapping vertices i and j.*

From the structural equivalence we define a stochastic block model over a partition of vertex sets.

Definition 7 (stochastic block model [192])**.** *Given a network $G = (V, E)$, or random adjacency matrix A, a tuple $(\mathcal{M}, p(A|\mathcal{M}))$ is a stochastic block model if*

(a) $\mathcal{M}$ is a set of pairwise disjoint subsets of V.

(b) With respect to p, we observe A_{ij} independently.

(c) With respect to p, vertices within the same block are stochastically equivalent.

We will repeatedly use capital K to denote number of blocks within which vertices share structural equivalence.

Definition 8 (size of a block model)**.** *Given a stochastic block model $(\mathcal{M}, p)$, the letter K denotes the number of blocks, that is $K = |\mathcal{M}|$.*

Mixed membership stochastic block models. A mixed membership stochastic block model relaxes the notion of blocks. Unlike a single membership block model partitions vertices into disjoint subsets, blocks in the mixed membership model may overlap with each other. In the original model [3], membership of a vertex to a certain block is fractional, and on each vertex sum of fractions equals to 1. In real-world network, where vertices may participate in multiple functional groups, this constraint is too strict and dilutes information. In recent studies the constraint was relaxed [94, 137], and the method was also extended to model dynamic networks [50].

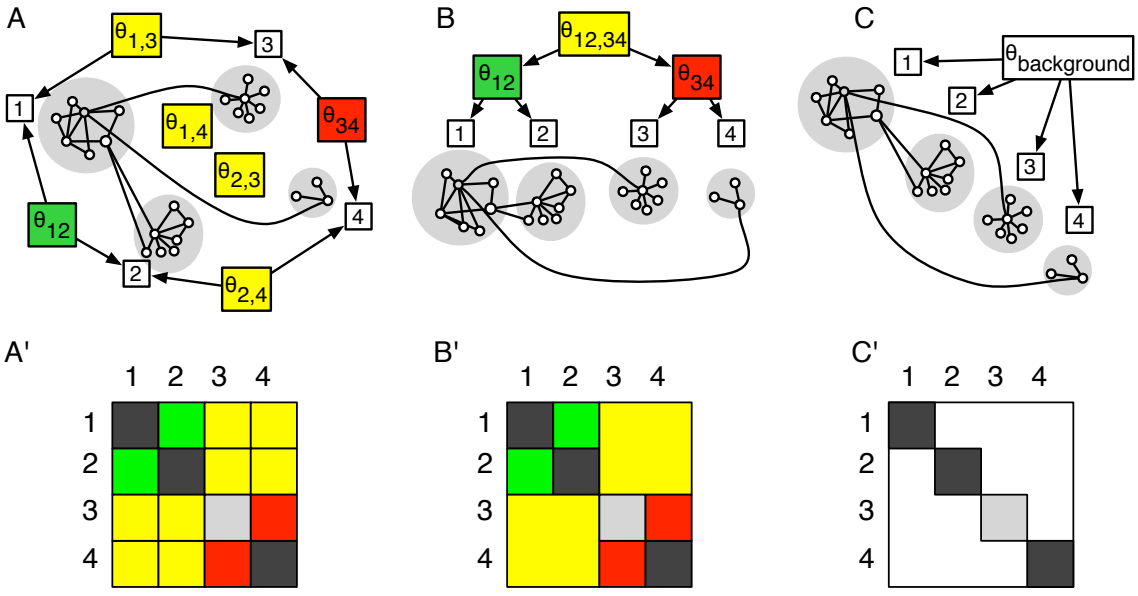

Figure 1.1: Examples of stochastic models. (A) A regular stochastic block model that represents all-pairwise block-block relations explicitly. (A′) The corresponding block matrix of the model A. (B) A hierarchical stochastic block model that share inter-block relations at a higher level. (B′) The corresponding block matrix of the model B. (C) A simplified stochastic block model that simplifies inter-block relations with a single parameter. (C′) The corresponding block matrix of the model C.

Hierarchical stochastic block model Hierarchical stochastic block models extend the regular stochastic block models, or the flat model. Unlike the flat model explicitly assign probabilities over all block-block pairs (Fig. 1.1A), the hierarchical model relationships between blocks are organized hierarchically and represented by a tree [29, 140, 142] (Fig. 1.1B). The hierarchical model reduces complexity, simplifying inter-block structures. For instance, at an intermediate level block-pairs between the left L blocks and right R blocks are parameterized by a single probability (e.g., top node of the tree in Fig. 1.1B), as opposed to $L \times R$ parameters/probabilities required by the flat block model.

The very first hierarchical stochastic block model was designed to branch exhaustively until bottom level blocks contain a single vertex [29]. It may be criticized that this type of hierarchical model takes complexity in the order of $O(n)$, whereas the all-pairwise block model takes $O(K^2)$ complexity. We know $n > K$ in general. In a small network, the hierarchical model may seem ineffective; however, the $O(n)$ model complexity increase much slower than $O(K^2)$ at a significantly large K. This makes the hierarchical model more generalizable than the flat block models.

It is possible that we may reduce complexity of the stochastic block model from the other perspective, completely ignoring the effect of inter-block relations. A simplest model could use just a single probability for all the edges occurring outside of blocks (Fig. 1.1C) [74]. But this drastic discount of complexity does not work well in real-world networks [133].

Bias variance tradeoff. We want to estimate a model from a certain class given the fact that observed network is noisy and incomplete. We conclude this section

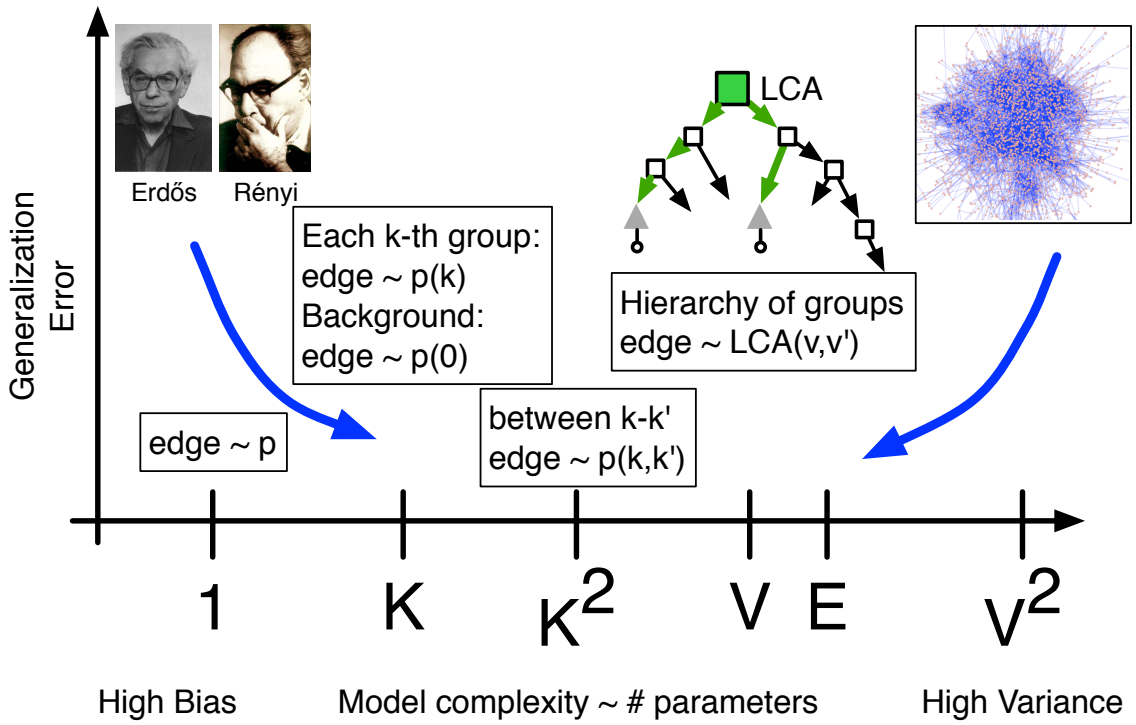

Figure 1.2: Bias-Variance tradeoff of variants of stochastic block models.

looking at various types of stochastic models in bias-variance tradeoffs (Fig. 1.2). We may begin modeling an observed network with the Erdös-Rényi (ER) model [41], and gradually increase a number of stochastically equivalent blocks, K, until the model of choice saddles on a minimum generalization error. The most complex model is indeed the observed network itself, or stochastic adjacency matrix with multiple observations on the same network [159]. Within the class of K-block models, model complexity varies from $O(K)$ to $O(K^2)$ depending on the choice of inter-block probabilities, a single background probability [74] to the all-pairwise flat block model [135].

Although the ER model may seem too simplistic, the model will provide a reasonable estimate of total edge density, can serve as a null model for hypothesis

testing [78,123], and Bayesian model comparison [140]; modeling a full $O(n^2)$ adjacency matrix may seem unrealistic, but will become tractable with dense sampling on overall pairs. One of our goals is to find most appropriate models given the amount of networks and quality of experiments.

1.4 Research questions

In this dissertation research we will seek answers to the following questions: How biological networks are organized? How networks change in the dynamic process? How should we take advantages of network clusters? To answer these questions, we face computationally challenging problems. First, given a plethora of models and methods we choose an appropriate model that describes networks most accurately; given a class of models, finding most probable model is also another challenge. We therefore developed efficient approaches to model fitting and model comparison (Chapters. 2 and 3). Next, we extended the static algorithms to dynamic clustering / matching algorithms, to be able to analyze dynamically changing networks (Chapter. 4). Developed methods revealed largely unknown mechanisms and dynamics in diverse biological systems. Finally, we address practical issues in network analysis (Chapter. 5). We proposed a promising solution to prioritization of network modules in subsequent researches, using discriminative learning. Our proposed approach can easily extend to "big data" analysis that could take full advantage of large databases.

Most chapters build upon the published works, Dynamic Hierarchical Model (DyHM) paper [142], Hierarchical Agglomerative Clustering (HAC) paper [140] and Dynamic Hierarchical Agglomerative Clustering (DHAC) paper [141]. All

of the contents in Chapter. 2 was based on the HAC paper [140]. Chapter. 3 was partially borrowed from the DyHM paper [142], but most of the results and the algorithms of degree-corrected models were unpublished yet. In Chapter. 4 the dynamic extension of statistical inference method was published in the DyHM paper [142], and dynamic agglomerative clustering and set matching were published in the DHAC paper [141]. All of the results and algorithms in Chapter. 5 are unpublished yet.

Chapter 2

Network modules by hierarchical agglomerative clustering algorithm

2.1 Introduction

A hierarchical network model [29], proposed by Clauset, Moore and Newman, provides a principled method for investigating structure at all levels by defining a probability distribution over network structures. However the original algorithm they proposed rely on lengthy Markov chain Monte Carlo (MCMC) simulation, which is very cumbersome for networks of more than 1,000 vertices and 10,000 edges. More fundamentally, this model imposes an exhaustive hierarchical structure at all levels of a network, even on cliques located at the very bottom level.

Here we propose a new approach, Hierarchical Agglomerative Clustering (HAC) that provides a scalable, deterministic approximation for optimizing a network probability motivated by CMN. The HAC was motivated by a key observation by Newman and Leicht [133], that interactions with vertices outside a group often provide more information than within-group interactions. Methods that focus on within-cluster interactions, such as modularity scores [30], Bayesian Hierarchical Clustering [70], and even spectral methods [114] often miss this infor-

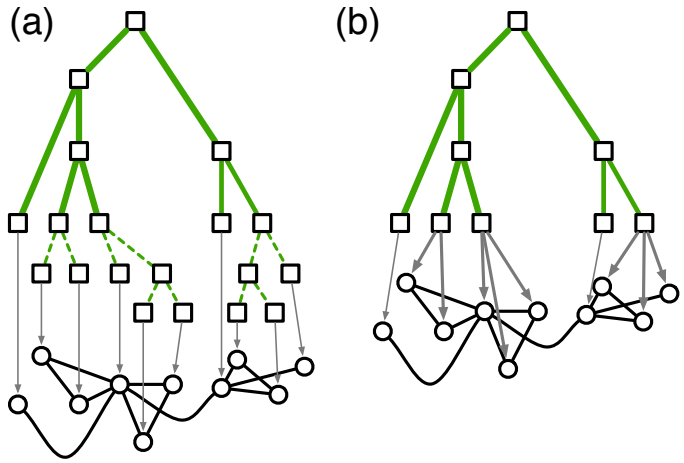

Figure 2.1: Hierarchical network models. (a) The original model proposed by Clauset *and coworkers* . represent underlying network data by exhaustively bisections. (b) We generalized the model, permitting non-informative sub-trees could be collapsed at the bottom level. Here we mark the collapsed sub-trees by dashed lines.

mation. We use both within- and between-cluster information to drive accurate bottom-up clustering using a novel model selection strategy to identify groups to merge and to detect when a subtree should be collapsed into a single cluster, similar to Power Graph [160] but with a firm statistical foundation. A similar Bayesian model selection step determines when clustering should be terminated, yielding a set of top-level clusters lacking evidence for further hierarchical structure.

2.2 Preliminary

A network G is defined by a set of vertices V and edges E that connect pairs of vertices. This work considers undirected, unweighted edges with no self-edges. Extensions to directed, weighted, and self-edges are possible but are not discussed here.

A "flat" model. A model $\mathcal{M}$ defines how vertices are collected into groups. These groups are denoted $C_1, C_2, \ldots, C_K$ for a model with K groups. Each vertex is assigned to one of the K groups, and the groups are disjoint. This model can be summarized as $\mathcal{M} = \{C_k : k \in 1, \ldots, K\}$. Subscripts u, v typically refer to individual vertices, and subscripts i, j, k refer to groups.

Edge counts between groups can be summarized as $e_{ij} = \sum_{u \in i, v \in j} e_{uv}$ for $i \neq j$, and $e_{ii} = \sum_{u < v \in i} e_{uv}$. The binary variable $e_{uv} = 1$ for a $u \sim v$ edge and 0 for the lack of an edge, or a hole. Total pair counts are defined as $t_{ij} = n_i n_j$ for $i \neq j$, and $t_{ii} = n_i(n_i - 1)/2$, where n_i is the number of vertices within group i. Summary counts for holes are $h_{ij} = t_{ij} - e_{ij}$. For a given pair of groups i and j, the e_{ij} edges are modeled as the result of t_{ij} independent Bernoulli trials with parameter θ_{ij}. The probability of the observed edges, conditioned on θ_{ij}, is

$$P_{ij}(\theta_{ij}) \;=\; \theta_{ij}^{e_{ij}} (1 - \theta_{ij})^{h_{ij}}. \tag{2.1}$$

The maximum likelihood value P_{ij}^{ML} is obtained by setting θ_{ij} to its maximum likelihood estimate with a uniform prior, $\hat{\theta}_{ij} = e_{ij}/t_{ij}$. A fully Bayesian probability P_{ij}^{FB} is obtained by integrating out the nuisance parameter θ_{ij}, again with a uniform prior:

$$\begin{aligned} P_{ij}^{ML} &\equiv e_{ij}^{e_{ij}} h_{ij}^{h_{ij}} / t_{ij}^{t_{ij}} \\ P_{ij}^{FB} &\equiv \text{Beta}(e_{ij} + 1, h_{ij} + 1) \end{aligned} \tag{2.2}$$

where Beta is the standard Beta function and $x^x = 1$ for $x = 0$.

For a flat model, with $K(K+1)/2$ parameters, the likelihood and fully Bayesian

probability are

$$\begin{aligned} \mathcal{L}(G;\mathcal{M}) &= \prod_{i\le j} P_{ij}^{ML}, \\ P(G|\mathcal{M}) &= \prod_{i\le j} P_{ij}^{FB}. \end{aligned} \tag{2.3}$$

Generalization to a hierarchical model. We can extend the notion of a model $\mathcal{M}$ to a hierarchical random graph (HRG) based on a model that successively merges pairs of groups [29]. This original model generates a binary dendrogram T. Each node r in this dendrogram represents the joining of network vertices $L(r)$ underneath the left sub-tree and vertices $R(r)$ underneath the right sub-tree. With the same Bernoulli probability model (Eq.2.1) as a building block, e_r and h_r are defined as the total number of edges and holes crossing between the left and right sub-trees. We generalize this model for the case of multiple top-level nodes, which merge together into a flat structure using a full block model. We also generalize for tree structures that are not completely branching, yielding tree nodes that collect multiple network vertices into a single group. Similar to Eq.2.3, letting $\mathcal{M} \equiv T$, the likelihood $\mathcal{L}(G;\mathcal{M})$ of a hierarchical model T and the corresponding probability $P(G|\mathcal{M})$ of the network given the model are

$$\begin{aligned} \mathcal{L}(G;\mathcal{M}) &= \prod_{r\le r'\in \text{top}} P_{rr'}^{ML} \prod_r P_r^{ML} \\ P(G|\mathcal{M}) &= \prod_{r\le r'\in \text{top}} P_{rr'}^{FB} \prod_r P_r^{FB}. \end{aligned} \tag{2.4}$$

Top-level terms $P_{rr'}^{ML} = e_{rr'}^{e_{rr'}} h_{rr'}^{h_{rr'}} / t_{rr'}^{t_{rr'}}$ and $P_{rr'}^{FB} = \text{Beta}(e_{rr'}+1, h_{rr'}+1)$ depend on the edges $e_{rr'}$ and holes $h_{rr'}$ crossing between the top-level groups r and r', with $t_{rr'} = e_{rr'} + h_{rr'}$. For all tree nodes, $P_r^{ML} = e_r^{e_r} h_r^{h_r} / t_r^{t_r}$ and $P_r^{FB} = \text{Beta}(e_r+1, h_r+1)$. For branching nodes (including the top-level nodes), the edges e_r holes h_r refer to

those crossing between the left and right sub-trees; for non-branching terminals, e_r and h_r refer to the edges and holes for vertices within the terminal groups.

2.3 Algorithm

Our approach is similar to Bayesian Hierarchical Clustering [70]. We start from a model $\mathcal{M}$ that clusters consist of a single vertex, and reduce model complexity by merging a best pair of clusters into a new larger cluster. In the course of iterative model comparison we build a hierarchical stochastic block model as a "byproduct." This hierarchical model embeds a wide spectrum of model complexity, more exactly a set of models at multiple resolution. We then choose a model that suits for our purpose. Two phases summarize the overall process: (1) building a guide tree; (2) collapsing the tree.

Maximum likelihood guide tree. Suppose currently there are K top-level clusters numbered $1 \dots K$ within the R total tree nodes. This model, $\mathcal{M}$, has $K(K-1)/2 + R$ total parameters. Merging two of the top-level nodes (and retaining the structure underneath each) gives a model with $(K-1)(K-2)/2 + (R+1)$ parameters, a reduction of $K-2$ parameters. Without loss of generality suppose we merge clusters 1 and 2 into a new cluster $1'$, defining a new model $\mathcal{M}'$. The model likelihood ratio is

$$\lambda_{12}^{ML} \equiv \frac{\mathcal{L}(\mathcal{M}')}{\mathcal{L}(\mathcal{M})} = \prod_{k=3}^{K} \frac{P_{1'k}^{ML}}{P_{1k}^{ML} P_{2k}^{ML}}. \tag{2.5}$$

There is a subtle but crucial difference between this agglomerative algorithm, which assumes a full block model for the top-level nodes, and the more standard approach with a star-like structure at the top with a single parameter governing

the interactions between all pairs of top-level nodes. A star-like model with K top-level and R total nodes has $R+1$ parameters, and merging two groups increases the number of parameters by 1. The increase in parameters at each step, coupled with a maximum likelihood model, is liable to over-fit the group structure. A further problem is the model likelihood ratio for the star model,

$$\begin{aligned}\lambda^{\star}_{12} &= \frac{e_{12}^{e_{12}} h_{12}^{h_{12}}}{t_{12}^{t_{12}}} \cdot \frac{t_b^{t_b}}{e_b^{e_b} h_b^{h_b}} \cdot \\ &\quad \frac{(e_b - e_{12})^{e_b - e_{12}} (h_b - h_{12})^{h_b - h_{12}}}{(t_b - t_{12})^{t_b - t_{12}}},\end{aligned} \tag{2.6}$$

where $e_b = \sum_{k<k'=1}^{K} e_{kk'}$ and similarly $h_b = t_b - e_b$ count the edges and holes between all pairs of top-level groups before merging 1 and 2, and e_{12} and h_{12} count the edges and holes just between groups 1 and 2. Under the star model, any two groups with the same values of e_{12} and t_{12} will have identical ratios $\lambda^{\star}_{12}$. At the initial step, every pair of vertices will have one of two merging scores, depending on whether $e_{12} = 1$ or 0. Additional criteria are then required to avoid bad merges at the start of clustering. In contrast, λ^{ML}_{12} gathers information from shared patterns of connectivity with other groups. In particular, at the initial step when each group is a single vertex, $\lambda^{ML}_{12} = (1/2)^{\#\text{mismatches}}$, where the number of mismatches is $\sum_{k=3}^{K} e_{1k} h_{2k} + h_{1k} e_{2k}$.

Greedy agglomerative algorithm. The likelihood ratio λ^{ML}_{12} leads to an agglomerative algorithm that successively merges the two clusters have the largest value. Alg. 1 summarizes overall steps. We call this method **HAC-ML**. The time complexity of a naïve implementation scales as $O(n^4)$, but using a priority queue, restricting possible merging pairs to clusters that share at least one common neighbor, and lazy evaluation of λ reduce the complexity to $O(mJ^2 \log n)$, where m is

the total number of edges and J is the average vertex degree, $\mathbb{E}[d_u]$.

Alg 1 HAC-ML

```
Initialize model M = {{v} : v ∈ V}
Initialize K ← V
while K > 1 do
    Find top-level clusters i,j with largest λ_ij^ML
    Add top-level cluster r; L(r) = i and R(r) = j
    Remove clusters i and j from the top level
    K ← K − 1
end while
```

Bayesian model selection for top- and bottom-level clusters. A natural stopping criteria at the top level is obtained by augmenting λ_{12}^{ML}, Eq. 2.5 with its fully Bayesian equivalent λ_{12}^{FB},

$$\phi_{12}^{FB} \equiv \prod_{k=3}^{K} \frac{P_{1'k}^{FB}}{P_{1k}^{FB} P_{2k}^{FB}}. \tag{2.7}$$

A reasonable stopping criterion is $\lambda_{ij}^{FB} \leq 1$ for the best merge [91]. While there are $K(K-1)/2$ possible merges, we do not include this factor in the stopping criterion.

Clusters with a single vertex are considered collapsed. During the merging process, if clusters 1 and 2 are selected for merging and are both collapsed, the probability ratio

$$\phi_{12}^{C} \equiv \frac{\text{Beta}(\sum_{i\leq j=1}^{2} e_{ij} + 1, \sum_{i\leq j=1}^{2} h_{ij} + 1)}{\prod_{i\leq j=1}^{2} \text{Beta}(e_{ij} + 1, h_{ij} + 1)}. \tag{2.8}$$

is calculated, where the subscripts indicate edges and holes within and between groups. The merged cluster is collapsed if $\lambda_{12}^{C} \geq 1$. Clusters of two vertices are always merged because $\lambda^{C} = 1$. While there are $2^{n_1+n_2} - 2$ ways for the reverse

process of splitting a cluster into two non-empty groups of sizes n_1 and n_2, we do not include this factor in the model selection.

Extension to multiple edge types. The HAC-ML algorithm is directly applicable to networks with multiple edge types. Rather than merging the edges into a single superimposed network, each edge type α defines its own likelihood $\mathcal{L}^{(\alpha)}(\mathcal{M})$ and probability $P^{(\alpha)}(\mathcal{M})$ for a particular model $\mathcal{M}$. The full likelihood and full probability are then obtained as products over the edge types, $\mathcal{L} = \prod_\alpha \mathcal{L}^{(\alpha)}$ and $P = \prod_\alpha P^{(\alpha)}$.

2.4 Results

Link prediction. We assessed correctness of a model in the framework of link prediction as presented in Hendersonand coworkers [71]. Starting with a real-world network, training networks are generated by deleting a specified fraction of edges. A test set is defined by the held-out edges and a random choice of an equal number of holes. This test set definition is suitable for assessment, but overstates practical performance by reducing the number of negative test examples for a sparse network. Note that for large real-world networks, group assignments are generally unknown, making it difficult to assess group assignments directly.

We then ran all methods on the training data set. The trained group structure provides maximum likelihood estimates for edges within and between clusters (Eq. 2.12). For VBM and CNM, we estimated edge densities between all pairs of clusters and within all clusters. For hierarchical models, we estimated densities between all left and right clusters at all tree levels. For GDK, each pair's diffusion

was directly used to rank pairs. Finally we assessed precision and recall of pairs in the test set ranked by link probability or GDK score.

Varying a cutoff c for similarity scores s_{uv} of a pair (u, v), we count a number of true positives (TP), false positives (FP), and false negatives (FN). In practice we determine levels of cutoff values by the decreasing order of scores.

$$\begin{aligned} \mathsf{TP} &= \mathbb{1}[s_{uv} \geq c \wedge e_{uv} = 1], \\ \mathsf{FP} &= \mathbb{1}[s_{uv} \geq c \wedge e_{uv} = 0], \\ \mathsf{FN} &= \mathbb{1}[s_{uv} < c \wedge e_{uv} = 1]. \end{aligned} \tag{2.9}$$

We draw a precision-recall curve (PRC) by

$$\mathsf{Precision} = \frac{\mathsf{TP}}{\mathsf{TP} + \mathsf{FP}} \quad \text{and} \quad \mathsf{Recall} = \frac{\mathsf{TP}}{\mathsf{TP} + \mathsf{FN}}. \tag{2.10}$$

Similarly, we define the true positive and false positive rates

$$\mathsf{TPR} = \frac{\mathsf{TP}}{\mathsf{TP} + \mathsf{FN}}, \quad \mathsf{FPR} = \frac{\mathsf{FP}}{\mathsf{FP} + \mathsf{TN}}, \tag{2.11}$$

which constitutes a receiver operating characteristic (ROC) curve. The F-score is the maximum value of harmonic mean of precision and recall.

Data preparation Interaction data was mainly taken from BioGRID [174] (version 2.0.61) for physical interactions within *S. cerevisiae*, *A. thaliana*, *C. elegans*, *D. melanogaster*, and *H. sapiens*. Synthetic lethal and synthetic fitness defect genetic interactions were taken for *S. cerevisiae*. Additional genetic interaction data sets were collected from genome-wide Synthetic Gene Array (SGA) [31] and diploid-based Synthetic Lethality Analysis on Microarray (dSLAM) [138]. The largest network in this study contains roughly 5000 vertices and up to 100,000 interactions (Table. 2.1).

We ignored redundant pairs within each type of network such that resulting networks were undirected and unweighted. We then iteratively removed isolated or degree-1 vertices, as these provide scant information for clustering. For other non-BioGRID genetic interaction datasets we filtered out positively weighted pairs and applied the same iterative removal. In joint-network analysis, we restricted attention to the common intersection of genes.

Other methods. We compared HAC-ML with other deterministic methods: Fast Modularity (**CNM**; Clauset *and coworkers* [30]), Variational Bayes Modularity (**VBM**; Hofman and Wiggins [74], and Graph Diffusion Kernel (**GDK**; Qi *and coworkers* [151]). CNM is an efficient algorithm that directly optimizes Newman modularity [132]. VBM simplifies network data to one intra- and one inter-community probability distribution. For GDK by discriminating between even-length and odd-length paths, Qi*and coworkers* [151] improved link prediction performance, particularly for disassorative (bipartite-like) networks. We used the odd parity kernel with the recommended damping parameter set to 1.0.

Different merging scores. In addition, we also considered agglomerative clustering based on heuristic merging scores: (1) edge density, ρ_e; (2) combined edge density and shared neighbor density, $\rho_e + \rho_s$; and (3) decomposed Newman modularity Q from CNM [132]. The edge and shared neighbor densities for merging clusters 1 and 2 are

$$\rho_e(1,2) \equiv \frac{e_{12}}{t_{12}}, \tag{2.12}$$

$$\rho_s(1,2) \equiv \frac{\sum_{u \notin i,j}(e_{1u} > 0)\mathrm{AND}(e_{2u} > 0)}{\sum_{u \notin i,j}(e_{1u} > 0)\mathrm{OR}(e_{2u} > 0)}. \tag{2.13}$$

The summations in $\rho_s(1,2)$ runs over all vertices u not in groups 1 or 2, and the logical functions evaluate to 1 and 0. The Newman modularity for merging groups 1 and 2 is

$$Q_{12} = \sum_{u\in 1}\sum_{v\in 2} e_{uv} - (d_u d_v/2E), \tag{2.14}$$

where d_u and d_v are vertex degrees and E is the total number of edges. This algorithm is essentially CNM, but retains the hierarchical structure defined by the merge order for link prediction (rather than predicting links based on the cut that maximizes modularity). Replacing λ_{12}^{ML} with ρ_e, $\rho_e + \rho_s$, and Q yields algorithms HAC-E, HAC-ES, and HAC-Q.

Results Summary results for link prediction demonstrate overall superior performance by HAC-ML (Table. 2.2). Of the 8 real-world networks, HAC-ML is top or tied for top in link prediction 6 times, followed by GDK for 2, CNM for 2, and VBM for 1. These summary results are for 7.5% of known edges held out, and supplemented with an equivalent number of holes selected at random as an 85/15 cross-validation set.

More detailed results are provided for two of the largest networks, Yeast-PPI physical interactions (Fig. 2.2A,B,C) and Yeast-GEN genetic interactions (Fig. 2.2D,E,F). The HAC-ML method dominates along the precision-recall curve, and also generally performs best over many fractions of left-out edges (Fig. 2.2B,C,E,F). The high-precision region of the HAC-ML prediction generally extends further than the other methods (Fig. 2.2A,D).

Among top-ranked pairs, the flat models CNM and VBM perform worse than the hierarchical models. The performance of CNM is improved to nearly the

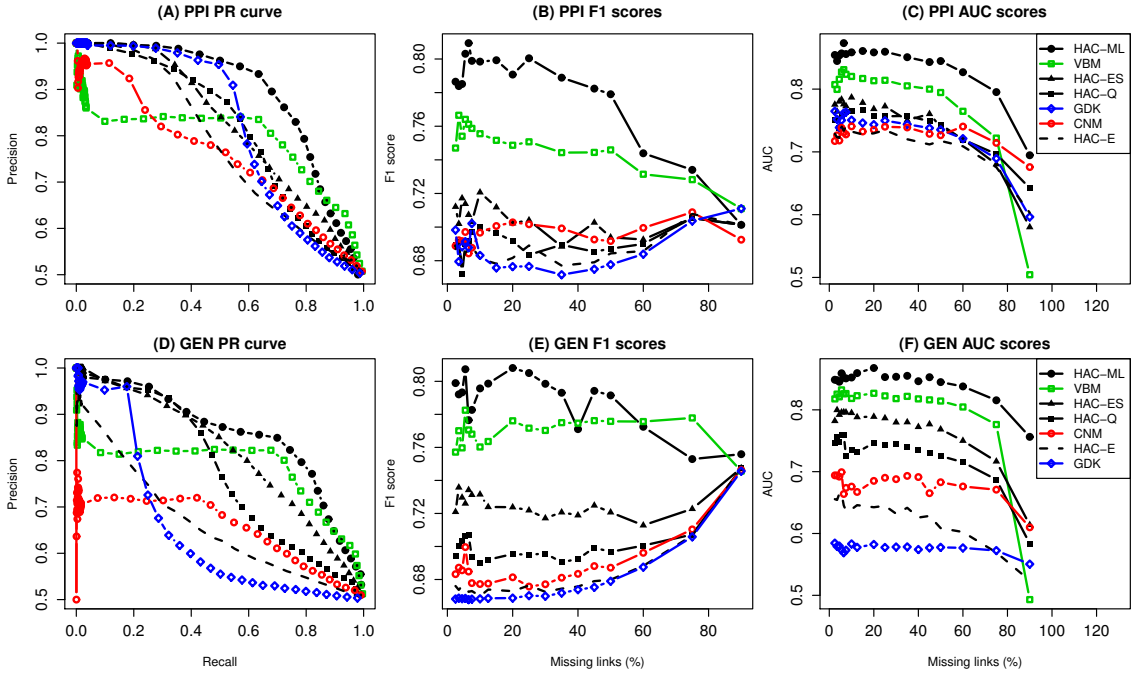

Figure 2.2: Link prediction results on Yeast networks. *A*: Precision Recall (PR) curve of 80/20 cross-validation experiment (CV) in YEAST-PPI dataset (10% missing links); *B*: F1 scores over different fractions of missing links in YEAST-PPI dataset from 1.5% to 90%; *C*: Area under ROC curve (AUC) scores over different fractions of missing links in YEAST-PPI dataset; *D*: PR curve of a 80/20 CV in YEAST-GEN dataset; *E*: F1 scores in YEAST-GEN dataset; *F*: AUC scores in YEAST-GEN dataset.

performance of HAC-ML by using HAC-Q to determine the merge order. The poor performance of CNM and VBM in the high-precision region may reflect the inherent resolution limit of a flat model [45] that hierarchical models do not appear to be limited.

Methods that consider shared neighbors, including HAC-ML and GDK, also perform better than methods that ignore this information, such as HAC-E. Shared neighbors are strong predictors of missing links in networks of protein interactions [58] and genetic interactions [198]. Methods that consider shared neigh-

bors, as opposed to just modularity or density, perform better for disassortative networks such as Yeast-GEN. The VBM method, which assumes homogeneous groups, may also work incorrectly when applied to networks with a mix of assortative and disassortative group structures.

Multi-resolution views of a physical interaction network A representative example of a top-level cluster with bottom-level structure is the protein transport complex discovered in the Yeast-PPI network (Fig. 2.3). This cluster, with 72 vertices, has a hierarchical structure with 4 layers branching down to over 10 bottom-level clusters. The bottom-level clusters include examples both of cliques (fully connected sets of vertices) and proteins that do not interact with each other but share common neighbors, including neighbors in other top-level groups.

Visual inspection indicates that the bottom-level clusters are subsets of known GO annotation categories, and may provide greater resolution than existing bottom-level GO categories. These results also indicate connections between GO categories learned from high-throughput data. An example is process of autophagy, which starts by forming a membrane-bound component that engulfs excess cytosolic proteins and make degraded in lysosome or other vacuoles [68,125]. Therefore "vesicle fusion" and "vesicle-mediated transport" are its mechanistic processes; a proper "protein localization" and targeting is required. Connections with plasma membrane proteins have become recently known, suggesting that plasma membrane is the source of autophagosome and *de novo* assembly of proteins and lipids initiate autophagy [33, 155]. As autophagy is a response to starvation [125] to re-use available intra-cellular resources. We find that disjoint low-level clusters correspond to "autophagy" and "golgi to plasma membrane trans-

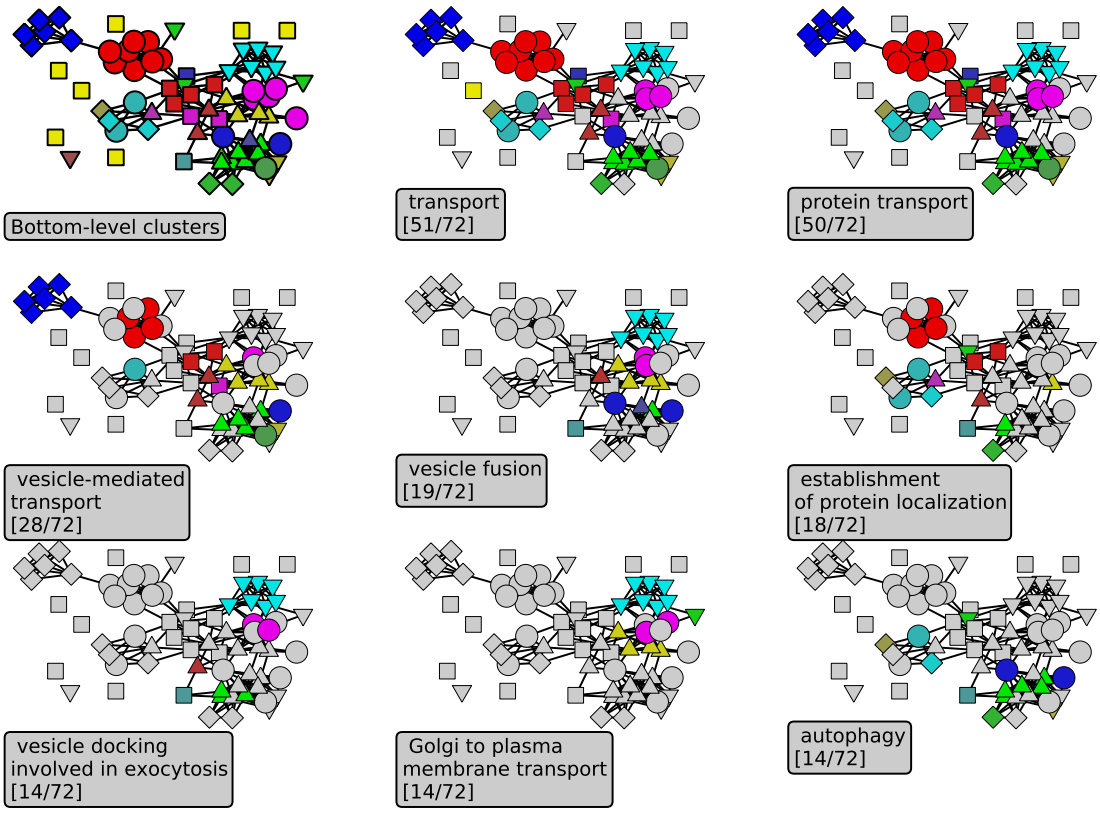

Figure 2.3: Protein transport complex. *Bottom level clusters*: Different shapes and colors in the topmost and leftmost panel indicate different bottom-level clusters; *Other panels*: Each box indicates one GO keyword and its enrichment within the subnetwork, and vertices belonging to this GO category are highlighted by non-gray colors.

port", suggesting that different proteins are responsible for transport in each direction. Moreover seemingly distant relationship to "exocytosis" is under investigation [148].

Synergy in mixed networks The extension to multiple edge types was used to compare link prediction for single yeast networks to link prediction from simultaneous analysis of physical and genetic interaction data (Table. 2.3). Little

evidence for synergy is apparent: predictions for a specific network are not improved by adding data from a second or third network. This behavior has been observed before for joint analysis of physical and genetic interactions [151, 152].

This lack of synergy may arise from high-throughput studies exploring different subsets of genes and proteins. Moreover our joint analysis assumes different types of edges are generated under a common group structure, but this pattern might be disrupted by a large fraction of false positive interactions, or some edge types might conflict with others. In presence of prevalent false positive interactions, physical and genetic interactions might not be *directly* complementary or orthogonal to each other in contrary to Kelley *and coworkers* [92]. In our simulation study, where orthogonality is well-preserved, HAC-ML trained by multiple data sources significantly outperformed (results not shown). To resolve this issue, a kernel-based method used by the previous studies [152] can be beneficial, but this is an open research problem.

Name	V	E	$\bar{d}$	Kind	Organism	Source
Arabidopsis	777	1,831	4.71	Physical	*A. Thaliana*	BioGRID[1]
Celegans	1,089	2,842	5.22	Physical	*C. elegans*	BioGRID[1]
Drosophila	4,692	19,876	8.47	Physical	*D. melanogaster*	BioGRID[1]
Human	6,094	26,112	8.57	Physical	*H. sapiens*	BioGRID[1]
Yeast-PPI	5,105	50,542	19.80	Physical	*S. cerevisiae*	BioGRID[1]
Yeast-GEN	4,763	85,855	36.05	Genetic	*S. cerevisiae*	BioGRID[1,2]
SGA	4,398	108,369	49.38	Genetic	*S. cerevisiae*	Costanzo *and coworkers* [3]
dSLAM	627	4,710	15.02	Genetic	*S. cerevisiae*	Pan *and coworkers* [4]

Table 2.1: Network data sets. *Symbols*: V, number of vertices (genes/proteins); E, number of edges (interactions); $\bar{d}$, average degree. *Data sources*: (1) BioGRID 2.0.61 [174]; (2) We selectively included "Negative Genetic", "Synthetic Growth Defect", "Synthetic Haploinsufficiency", "Synthetic Lethality" experiments; (3) Supp. Data S4, intermediate cutoff, of Costanzo *and coworkers* [31]; (4) Supp. Table S1 of Pan *and coworkers* [138].

Physical interactions							
Data	HAC-ML	GDK	CNM	VBM	HAC-ES	HAC-E	HAC-Q
Yeast-PPI	**0.79**±0.5	0.69±0.3	0.69±0.7	0.76±0.4	0.71±0.5	0.69±0.7	0.69±0.8
Drosophila	**0.73**±0.8	0.66±0.2	0.67±0.4	0.70±0.4	0.67±0.3	0.67±0.3	0.67±0.4
Human	0.73±0.9	**0.75**±0.7	0.71±0.5	0.70±0.6	0.67±0.4	0.68±0.5	0.69±1.0
Celegans	**0.68**±1.5	0.67±1.3	**0.68**±1.3	0.66±0.6	0.66±0.8	0.66±0.7	0.67±0.8
Arabidopsis	0.80±8.3	**0.92**±2.2	**0.92**±3.2	0.90±3.6	0.78±11.0	0.87±10.8	0.88±11.4
Genetic interactions							
Data	HAC-ML	GDK	CNM	VBM	HAC-ES	HAC-E	HAC-Q
Yeast-GEN	**0.78**±2.3	0.67±0.0	0.69±0.7	0.74±6.0	0.73±0.8	0.67±0.1	0.69±0.7
SGA	**0.76**±1.5	0.67±0.0	0.67±0.2	**0.76**±0.3	0.70±0.2	0.67±0.0	0.69±0.2
SLAM	**0.92**±1.0	0.91±0.5	0.68±0.8	0.67±0.3	0.84±2.9	0.76±1.0	0.67±0.3

Table 2.2: Link prediction performance of 85/15 cross validation (7.5% of observed edges held out). First numbers indicate an average F_1 score of multiple experiments and second numbers following $\pm$ sign are standard deviations of last-digit (multiplied by 100).

HAC-ML Trained by	Prediction of PPI	SGA	GEN
PPI	**0.75**±1.6		
SGA		**0.77**±1.0	
GEN			0.78±1.4
PPI+SGA	0.69±0.5	0.73±0.8	
PPI+GEN	0.71±1.1		**0.79**±0.5
SGA+GEN		**0.77**±1.0	0.78±1.1
PPI+SGA+GEN	0.68±1.2	0.73±0.3	0.78±0.6

Table 2.3: Link prediction performance of joint analysis. Evaluation scheme was 85/15 cross-validation. First numbers indicate an average F_1 score of multiple experiments and second numbers following $\pm$ sign are standard deviations of last-digit (multiplied by 100).

2.5 Biological impact

In this study we showed that our HAC-ML consistently outperformed other network clustering algorithms across all biological networks in cross-validation experiments. Cross-validation error is an objective measure when the gold standard model is unavailable; in some sense, our result implies that the models discovered by HAC-ML are closet to the unknown gold standard.

Our method can easily fit in to one component of a bioinformatic pipeline. Since quality of downstream analysis highly depends on quality of the upstream, high quality modular structure resolved by the HAC-ML will greatly enhance the accuracy of subsequent analysis. Recently Ideker *and coworkers* used our HAC-ML to increase coverage of conventional gene ontology graph [37].

What distinguishes the HAC-ML from others is the underlying model, which embeds a hierarchical and modular structure of vertices. Our experimental result is an empirical proof of long-standing conjecture on the existence of func-

tional modules [66]. Hierarchically organized modularity helps researchers narrow down hypothesis space. We reduced virtually n multiple hypotheses down to K, where n is number of genes and K is number of hierarchical modules and $n \gg K$. Moreover, the hierarchical structure permits efficient divide-and-conquer strategy.

2.6 Technical impact

Our hierarchical agglomerative clustering method works effectively and efficiently in real-world networks, with the ability to resolve both top-level and bottom-level groups. It provides superior performance for link prediction when applied to real-world networks, with a good tradeoff between efficiency and accuracy.

Unlike many agglomerative algorithms, which introduce a new parameter every time two groups are merged, HAC starts from a full model and removes parameters at each step. This approach gathers information from shared interaction patterns in building a guide tree, and then uses Bayesian model selection to collapse the bottom level of the tree and terminate the clustering at the top level.

Our initial attempt was made to use the Bayes factor for both guide tree building and collapsing. A problem with this approach is that the Bayesian likelihood includes a contribution, asymptotically the Bayes Information Criterion (BIC) correction [167], which favors merges of larger clusters with different connectivity patterns over merges of smaller clusters with identical connectivity patterns. Consequently, using the Bayesian likelihood optimized the local Bayes factor but gave a worse global Bayes factor than the maximum likelihood approach, which also has less expensive function evaluations. We therefore used maximum likeli-

hood for the guide tree and Bayesian likelihood for collapsing.

We can pose prior probability $P(\mathcal{M})$ over the model, i.e., a set of vertices. The probability that a vertex is in cluster k is π_k, the parameter for the k^{th} cluster in a multinomial distribution, with $\sum_k \pi_k = 1$. The model $\mathcal{M}$ generates a network G by first sampling the membership of each vertex u with probability π_k for cluster k, then sampling each edge $e_{uv} = 0$ or 1 as a Bernoulli trial with success probability θ_{ij} for $u \in i$ and $v \in j$.

A general weakness of deterministic optimization heuristics is possibility of becoming trapped in a local minimum. A more fundamental weakness is that different aspects of cross-cutting network structure may be reflected by multiple pertinent local minima. Even so, the group structure generated by HAC-ML can be used as a starting point for MCMC sampling over tree structures, which can provide better results than any single tree [29].

Chapter 3

Network modules by variational inference of a fixed hierarchical model

3.1 Introduction

Maximum likelihood estimation of the optimal tree (the optimal assignment of graph vertices to terminal leaves) is challenging since it involves learning most likely left-right divisions for each parameter estimation task. The problem is similar to learning evolutionary parameters from an unknown phylogenetic tree structure. Related phylogeny algorithms escape this obstacle by performing Bayesian model averaging rather than attempting to identify the optimal model. For example, the Metropolis-Hastings algorithm [67] can sample plausible tree structures according to the likelihood; then, based on the ensemble of these trees, evolutionary parameters such as mutation rates can be estimated [99].

The previous work [29] uses model averaging by sampling over trees with probabilities obtained from maximum likelihood parameter estimates. In practice, this strategy is suitable for moderately small networks, and the model asymptotically converges to the Gibbs distribution of probable hierarchical structures, with probability proportional to their likelihood. Unfortunately, convergence can be difficult to determine, and adequate sampling can require substantial CPU re-

sources for even moderately sized networks (100 to 1000 vertices).

Our greedy HAC algorithm [140, 141] provides an excellent deterministic approximate of hierarchical group structure of a single network or multiple networks. We have seen the score function we used is statistical consistent. Nonetheless a greedy agglomerative algorithm is fundamentally irreversible, and errors occurring in first few merging steps are not fixable. Real-world networks are collected from noisy observational process. There is always a high chance of making mistakes.

Moreover, we have seen that total running time scale in the order of $O(md^2 \log n)$, with n number of vertices, m number of edges and d average vertex degree. In this respect, the HAC algorithm may not always be as scalable as desired. If d^2 grows as n or m, then it needs essentially $O(n^2)$ operations. Indeed an empirical study shows that d can grow as n, so called "the densification power-law" [107], and any algorithm scaling in the order of $O(n^2)$ has little practical usage on a large network of $10,000$ vertices.

3.2 Model definition

Structural approximation We consider another approximation approach based on probabilistic graphical model inference. First we fix the depth of the leaf nodes, and the dendrogram structure is a perfect binary tree. This approximation seems very restrictive, but has enough expressive power as long as we learn deep tree structure. For instance we may restrict the original hierarchical random graph in two steps: collapse unnecessarily branching subtrees; but allow a sufficiently larger fixed tree that contains the collapsed tree (Fig. 3.1 from a to c).

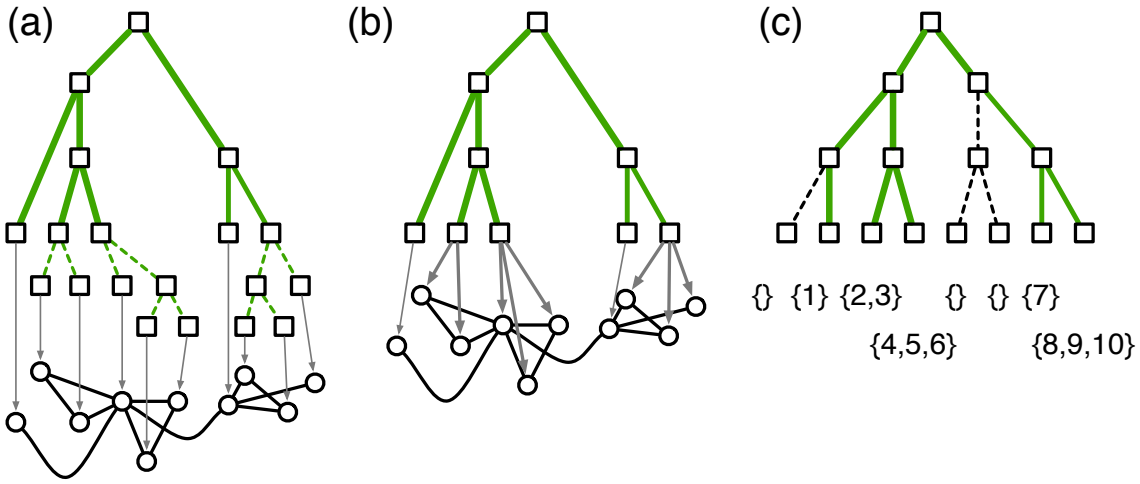

Figure 3.1: Structural approximation. (a) A fully branching binary dendrogram that represents hierarchical group structure of the network of 10 vertices and 10 edges. Each leaf node of the tree corresponds to a single network vertex. (b) A collapsed model that each leaf node corresponds to a set network vertices. (c) A fixed perfect binary tree that contains structure of the collapsed tree.

This structural assumption not only brings about a fixed probabilistic framework, which suits a variational approximation, but also reduces the search space from $O(n!!)$ to $O(K^n)$, where !! is the double factorial, $K = 2^{\text{depth}}$ is the number of terminal nodes, and n is the number of network vertices. As described in the results, this fixed dendrogram does not appear to change the results for occupied terminals provided that the tree is sufficiently deep, which is readily tested by runs at multiple tree depths.

A fixed binary tree model Let $G = (V, E)$ be a network data or graph, consisting of a set of vertices V and edges E. Indexes i, j typically denote index of network vertices and e_{ij} an undirected edge weight observed between vertices i and j. We define a fixed binary tree model $\mathcal{T}$ over the hierarchical partitions of vertices V. Not to obfuscate readers, we distinguish that an entity of the tree

model is node, not a vertex.

A fixed binary tree model $\mathcal{T} = (N, \mathcal{C}, \Theta)$ is defined by a set of nodes, a set of clusters $\mathcal{C} = \{C_k : k \in [K]\}$ and parameters $\Theta = \{\theta_k : k \in [K]\}$. Each terminal node k, or a node at the leaf-level, corresponds to a set of graph vertices, or a cluster C_k (Fig. 3.1c). We assume $C_k \cap C_{k'} = \emptyset$ for $k \neq k'$, and $\bigcup_{k\in[K]} C_k = V$. Each internal node r divides terminal nodes $[K]$ into the left and right, which we denote respectively L_r and R_r. Consequently the node r divides a subset of network vertices into left and right, that is $V_{L_r} = \bigcup_{k\in L_r} C_k$ and $V_{R_r} = \bigcup_{k\in R_r} C_k$.

Tree nodes are also associated with probability distributions; for each node r we assign a unique parameter θ_r, which then characterizes distribution of sub-network observed underneath r. For a terminal node k, we have a sub-network $G' = (V_k, E_k)$. Obviously its vertex set is same as the cluster C_k, i.e., $V_k = C_k$ and the edge set $E_k \subset E$ contains edges whose both endpoints belong to V_k, i.e., $E_k = \{e_{ij} \in E : i, j \in V_k\}$. In the model a probability $p(E_k|\theta_k)$ parameterized by θ_k describe the density of observed set E_k. At an internal node r the model $\mathcal{T}$ defines the probability of edges E_r between the left V_{L_r} and right V_{R_r}, that is $E_r = \{e_{ij} \in E : i \in V_{L_r} \wedge j \in V_{R_r}\}$.

For convenience we let n_k count number of vertices and m_k number of edges underneath a terminal node k. In other words, $n_k = |V_k|$ and $m_k = \sum_{e\in E_k} e$. Likewise we let

$$n_{L_r} = |L_r|, \quad n_{R_r} = |R_r| \quad \text{and} \quad m_r = \sum_{e\in E_r} e.$$

For example, the likelihood function of original hierarchical random graph [29] is

$$\mathcal{L}(G; \mathcal{T}) = \prod_{r\in N} \theta_r^{m_r} (1 - \theta_r)^{n_{L_r} n_{R_r} - m_r}, \tag{3.1}$$

while the likelihood of HAC [140] is

$$\prod_{r\in\text{internal}} \theta_r^{m_r}(1-\theta_r)^{n_{L_r}n_{R_r}-m_r} \prod_{k\in\text{terminal}} \theta_k^{m_k}(1-\theta_k)^{n_k(n_k-1)/2-m_k}. \tag{3.2}$$

A latent variable model We can reformulate the same model in terms of latent variables. Let $\mathcal{T} = (N, Z, \Theta)$. N and Θ are the same, a set of nodes and parameters, but Z is a $n \times K$ matrix. An element z_{ik} is a latent variable indicating whether vertex i is assigned to the terminal node k: $z_{ik} = 1$ only if i^{th} vertex is assigned to that node, otherwise $z_{ik} = 0$.

For succinctness we introduce another notation, lowest common ancestor LCA in the tree structure. We let $\text{LCA}(\text{a}, \text{b})$ be an internal node which two paths from terminal nodes a and $b \in [K]$ meet first in bottom-up traversal. If $a = b$, obviously $\text{LCA}(\text{a}, \text{b}) = \text{a}$. Using this, we count number of edges at a terminal k and an internal node r, respectively

$$m_k = \sum_{i<j} e_{ij} z_{ik} z_{jk} \quad \text{and} \quad m_r = \sum_{i,j} \sum_{a,b} e_{ij} z_{ia} z_{jb} \mathbb{1}[\text{LCA}(\text{a}, \text{b}) = \text{r}]. \tag{3.3}$$

Similarly we get the number of vertices on the left and right of an internal node r

$$n_{L_r} = \sum_{a\in L_r} \sum_{i=1}^{n} z_{ia} \quad \text{and} \quad n_{R_r} = \sum_{b\in R_r} \sum_{i=1}^{n} z_{ib}; \tag{3.4}$$

the size of bottom-level cluster k is simply

$$n_k = \sum_{i=1}^{n} z_{ik}. \tag{3.5}$$

Hierarchical stochastic block model Now consider the latent variable model from the perspective of vertices. Suppose i and j belong to bottom-level cluster a and b respectively, we observe an edge, $e_{ij} = 1$, with probability $\theta_{\text{LCA}(\text{a},\text{b})}$. We

define the likelihood function,

$$\mathcal{L}(G;\mathcal{T}) = \prod_{a,b}\prod_{i<j} P(e_{ij}|z_{ia}=1,z_{jb}=1,\Theta)^{z_{ia}z_{jb}}, \quad (3.6)$$

$$P(e_{ij}|z_{ia}=1,z_{jb}=1,\Theta) = \theta_{\mathrm{LCA(a,b)}}^{e_{ij}}(1-\theta_{\mathrm{LCA(a,b)}})^{1-e_{ij}}. \quad (3.7)$$

We assumed a non-informative prior on θ, i.e., $\theta_r \sim \mathsf{Beta}(1,1)$.

Degree correction In many real-world networks, including biological networks [82], we have seen emergence of high-degree vertices, or hubs, which is characterized by a heavy-tailed degree distribution [14]. Over the last decade systems biology community have tried to understand a fundamental role of hubs in regulatory contexts [1,16,17,27,64,96,180].

A degree-corrected stochastic block model [89] [1] accounts for degree sequences observed in network data. A main idea is to incorporate expected potential of connectivity between two vertices into model parameters, where the expectation is calculated under observed degree sequences. In the generative scheme, there are two sources of uncertainty one from degree sequences; the other determined by a block, or a cluster, to which two endpoints belong.

Suppose we have a sequence of vertex degrees $\{d_i : i \in [n]\}$ of n vertices. Then we expect vertices i and j form an edge with the null probability

$$\rho_{ij} = \mathbb{E}\left[e_{ij}\right] = \frac{d_i d_j}{2m} \quad (3.8)$$

because each vertex i can be chosen with probability $d_i/2m$ and we repeat the selection process $2m$ times. If the vertex i belongs to bottom level a and j to b, i.e., $z_{ia}=1$ and $z_{jb}=1$, then we observe e_{ij} in two stages of events sampling from $P(e_{ij}|\theta_{\mathrm{LCA(a,b)}})$ and $P(e_{ij}|\rho_{ij})$.

[1] Petterson and Bader (*unpublished*) first introduced the same idea.

In our degree-corrected model [89], we assume these two events occur independently and multiplicative: we observe e_{ij} sampled from a Poisson distribution with the rate parameter $\theta_{\text{LCA(a,b)}} \cdot \rho_{ij}$. The full likelihood function is

$$\mathcal{L}(G;\mathcal{T}) = \prod_{a,b}\prod_{i<j} P(e_{ij}|z_{ia}=1, z_{jb}=1, \Theta)^{z_{ia}z_{jb}}, \tag{3.9}$$

$$P(e_{ij}|z_{ia}=1, z_{jb}=1, \Theta) = \text{Pois}\Big(e_{ij}\Big|\theta_{\text{LCA(a,b)}}\rho_{ij}\Big) \tag{3.10}$$

$$= \frac{1}{e_{ij}!}(\theta_{\text{LCA(a,b)}}\rho_{ij})^{e_{ij}} e^{-\theta_{\text{LCA(a,b)}}\rho_{ij}}. \tag{3.11}$$

We model the prior distribution on θ follows the Gamma distribution.

3.3 Bayesian inference

A Bayesian inference algorithm estimates posterior the probability of unknown model given observed data set. Our observed data $\mathcal{D}$ is a set of edges (and weights), $\mathcal{D} = \{e_{ij}\}$; latent variable matrix Z and block parameters Θ constitute the unknown model $\mathcal{T}$. Therefore our goal is to estimate $P(Z|\mathcal{D})$ and $P(\Theta|\mathcal{D})$.

Variational approximation However, even after the structural approximation full and exact posterior computation is intractable, since the number of possible latent states scales in K^n with K number of bottom-levels and n number of vertices. A full Markov chain Monte Carlo simulation is ineffective in a network with $n > 1000$. Here we use variational approximation method [84] that can find approximate distribution of unknown random variables in deterministic optimization. More explicitly, we will find a surrogate distribution $q(Z, \Theta)$ that approximates

$$P(Z, \Theta|G) \propto \prod_{a,b}\prod_{i,j} P(e_{ij}|\theta_{\text{LCA(a,b)}})^{z_{ia}z_{jb}} \prod_{r\in[N]} P(\theta_r). \tag{3.12}$$

We use fully factored q to ease computations:

$$Q(Z, \theta|\mu, \alpha, \beta) = \prod_{i=1}^{n} Q(\mathbf{z}_i|\mu_i) \prod_{r\in\mathcal{T}} Q(\theta_r|\alpha_r, \beta_r), \tag{3.13}$$

where

$$Q(\mathbf{z}_i|\mu_i) = \prod_{k=1}^{K} \mu_{ia}^{z_{ia}},$$

and

$$\begin{aligned} Q(\theta_r|\alpha_r, \beta_r) &= \mathsf{Gam}(\theta_r|\alpha_r, \beta_r) \\ &= \frac{\beta_r^{\alpha_r}}{\Gamma(\alpha_r)} \theta_r^{\alpha_r - 1} e^{-\theta_r \beta_r} \end{aligned}$$

for the regular hierarchical model (Eq. 3.6);

$$\begin{aligned} Q(\theta_r|\alpha_r, \beta_r) &= \mathsf{Beta}(\theta_r|\alpha_r, \beta_r) \\ &= \frac{\Gamma(\alpha_r + \beta_r)}{\Gamma(\alpha_r)\Gamma(\beta_r)} \theta_r^{\alpha_r - 1} (1 - \theta_r)^{\beta_r - 1} \end{aligned}$$

for the degree-corrected model (Eq. 3.9).

Coordinate ascent algorithm

Mean-field approximation We find a variational distribution (Eq. 3.13) that is closest to the actual posterior distribution (Eq. 3.12). To measure closeness or distance we use Kullback-Leibler (KL) divergence, that is

$$D_{\mathrm{KL}}(f\|g) = \int dx \, \log \frac{f(x)}{g(x)} f(x), \tag{3.14}$$

where f and g are a probability density function. Note the KL divergence is asymmetric; i.e., generally $D_{\mathrm{KL}}(P\|Q) \neq D_{\mathrm{KL}}(Q\|P)$.

In mean-field approximation we find Q that minimizes

$$D_{\mathrm{KL}}(Q\|P) = \mathbb{E}_Q[\log Q] - \mathbb{E}_Q[\log P]\,. \tag{3.15}$$

Since the mean-field distribution Q (Eq. 3.13) is fully factorized this side of KL divergence greatly simplifies computations. The convexity of KL divergence guarantees that we can find an optimum Q by taking coordinate steps: we optimize each factorized component of Q (eq. 3.13) one by one until convergence. For an exponential family distribution, as in our models, the coordinate steps can be further simplified. A general theory suggests this optimization is equivalent to derive a posterior probability of each random variable conditioned on dependent potential functions–terms inside of the exponent [196]. For a more precise and explicit notion of dependency we refer readers to the original article [196] or a general text [95].

There are two types of random variables in our model: the latent variables z_{ia} for a vertex i and bottom-level cluster a, and the parameters θ_r for tree nodes. We update the former locally and the latter globally [73] since each latent variable associates local membership of the vertex but tree parameters can only be determined by sufficient statistics that require all the latent membership of vertices.

We derive the update equations based on this generalized mean-field theory [196]. For each random variable, we first find the posterior conditioned on other dependent variables, then formulate it in an exponential form, and remove uncertainty of potential functions taking expectation with respect to the appropriate variational distribution.

Local update by mean-field Let us characterize the local distribution $Q(z_{ia}|\cdot)$, assignment of a vertex i to bottom-level a. It is straightforward to find the poste-

rior z_{ia} given Θ and other $\mathbf{z}_j$ ($j \neq i$):

$$P(z_{ia} = 1|\cdot) \propto \prod_{r\in[N]} \prod_{b\in[K]:\, r=\mathrm{LCA(a,b)}} \prod_{j\neq i} P(e_{ij}|z_{ia} = 1, z_{jb}, \theta_r).$$

In an exponential form

$$\prod_{j\neq i} P(e_{ij}|z_{ia}, z_{jb}, \theta_r) \propto \exp\left(\eta_r^\top \mathbf{s}_{ib}\right),$$

we have

$$\eta_r = \left(\log\frac{\theta_r}{1-\theta_r}, \log(1-\theta_r)\right)^\top, \quad \mathbf{s}_{ib} = \left(\sum_j z_{jb}e_{ij}, \sum_j z_{jb}\right) \tag{3.16}$$

for the uncorrected block model (Eq. 3.6);

$$\eta_r = (\log\theta_r, -\theta_r)^\top, \quad \mathbf{s}_{ib} = \left(\sum_j z_{jb}e_{ij}, \sum_j z_{jb}\rho_{ij}\right) \tag{3.17}$$

for the degree corrected model (Eq. 3.9). Removing the uncertainty of $\eta_r(\theta_r)$, we have

$$Q(z_{ia} = 1|\cdot) \propto \prod_{r\in[N]} \exp\left(\mathbb{E}_Q[\eta_r]^\top \sum_{b\in[K]:\, r=\mathrm{LCA(a,b)}} \mathbb{E}_Q[\mathbf{s}_{ib}]\right). \tag{3.18}$$

Local update by collapsed variational inference Recently a slightly different type of local update was introduced [190], which we term locally collapsed variational inference, or LCVI. Loosely speaking, LCVI updates minimize the other side of KL divergence,

$$D_{\mathrm{KL}}(P\|Q) = \mathbb{E}_P[\log P] - \mathbb{E}_P[\log Q]. \tag{3.19}$$

In mean-field approximation we replace unknown global parameters, $\eta_r(\theta_r)$, with expected factors, $\mathbb{E}_{Q(\theta_r)}[\eta_r]$ (Eq. 3.18), but LCVI integrates out the unknown with

respect to the variational distribution,

$$Q(z_{ia}=1|\cdot) \propto \prod_{r\in[N]} \prod_{b\in[K]:\, r=\mathrm{LCA(a,b)}} \mathbb{E}_Q\left[\exp\left(\eta_r^\top \mathbb{E}[\mathbf{s}_{ib}]\right)\right]. \tag{3.20}$$

More specifically, for the uncorrected block model

$$\begin{aligned}\mathbb{E}_Q\left[\exp\left(\eta_r^\top \mathbb{E}[\mathbf{s}_{ib}]\right)\right] &\propto \mathsf{Beta}(a_r,b_r)^{-1}\int d\theta_r\, \theta_r^{a_r+\sum_j e_{ij}\mathbb{E}[z_{jb}]-1}(1-\theta_r)^{b_r+\sum_j \mathbb{E}[z_{jb}]-1}\\ &\propto \frac{\mathsf{Beta}\left(a_r+\sum_j e_{ij}\mathbb{E}\left[z_{jb}\right], b_r+\sum_j \mathbb{E}\left[z_{jb}\right]\right)}{\mathsf{Beta}(a_r,b_r)}\end{aligned} \tag{3.21}$$

where $\mathsf{Beta}(x,y)=\Gamma(x)\Gamma(y)/\Gamma(x+y)$; for the degree-corrected model,

$$\begin{aligned}\mathbb{E}_Q\left[\exp\left(\eta_r^\top \mathbb{E}[\mathbf{s}_{ib}]\right)\right] &\propto \frac{b_r^{a_r}}{\Gamma(a_r)}\int d\theta_r\, \theta_r^{a_r+\sum_j e_{ij}\mathbb{E}[z_{jb}]-1} e^{-(b_r+\sum_j \rho_{ij}\mathbb{E}[z_{jb}])\theta_r}\\ &\propto \frac{\Gamma(a_r+\sum_j e_{ij}\mathbb{E}\left[z_{jb}\right])b_r^{a_r}}{\Gamma(a_r)(b_r+\sum_j \rho_{ij}\mathbb{E}\left[z_{jb}\right])^{a_r+\sum_j e_{ij}\mathbb{E}[z_{jb}]}}.\end{aligned} \tag{3.22}$$

Global update We derive the global $Q(\theta_r|\cdot)$ similarly. For the uncorrected block model

$$Q(\theta_r|\cdot) \propto \exp\big((\mathbb{E}_Q[m_r]+a_0-1)\log(\theta_r)+(\mathbb{E}_Q[h_r]+b_0-1)\log(1-\theta_r)\big)$$

where $h_r = n_{L_r}n_{R_r}-m_r$ for an internal node r; and $h_k = n_k(n_k-1)/2$ for a terminal node k. We can characterize that

$$\begin{aligned}\theta_r &\sim \mathsf{Beta}(\theta_r|a_r,b_r), \quad \text{where} \\ a_r &\leftarrow a_0+\begin{cases}\sum_{i,j}\sum_{a\in L_r}\sum_{b\in R_r} e_{ij}\mathbb{E}[z_{ia}]\,\mathbb{E}\left[z_{jb}\right], & r \text{ is internal},\\ 0.5\sum_i \mathbb{E}[z_{ik}]\sum_{j\neq i} e_{ij}\mathbb{E}\left[z_{jk}\right], & r \text{ is terminal},\end{cases}\\ b_r &\leftarrow b_0+\begin{cases}\sum_{i,j}\sum_{a\in L_r}\sum_{b\in R_r} \mathbb{E}[z_{ia}]\,\mathbb{E}\left[z_{jb}\right], & r \text{ is internal},\\ 0.5\sum_i \mathbb{E}[z_{ik}]\sum_{j\neq i} \mathbb{E}\left[z_{jk}\right], & r \text{ is terminal}.\end{cases}\end{aligned} \tag{3.23}$$

For the degree-corrected model

$$Q(\theta_r|\cdot) \propto \exp\big((\mathbb{E}_Q[m_r] + a_0 - 1)\log\theta_r - (\mathbb{E}_Q[t_r] + b_0)\theta_r\big)$$

where $t_r = \sum_{i,j}\sum_{a,b}\rho_{ij}z_{ia}z_{jb}$ for an internal node r; and $t_k = \sum_{i<j}\rho_{ij}z_{ik}z_{jk}$ for a terminal node k. Therefore we have

$$\begin{aligned}
\theta_r &\sim \mathsf{Gam}(\theta_r|a_r, b_r) \quad \text{where} \\
a_r &\leftarrow a_0 + \begin{cases} \sum_{i,j}\sum_{a\in L_r}\sum_{b\in R_r} e_{ij}\mathbb{E}[z_{ia}]\,\mathbb{E}\left[z_{jb}\right], & r \text{ is internal,} \\ 0.5\sum_i \mathbb{E}[z_{ik}]\sum_j e_{ij}\mathbb{E}\left[z_{jk}\right], & r \text{ is terminal,} \end{cases} \\
b_r &\leftarrow b_0 + \begin{cases} \sum_{i,j}\sum_{a\in L_r}\sum_{b\in R_r} \mathbb{E}[z_{ia}]\,\mathbb{E}\left[z_{jb}\right] d_i d_j/2m, & r \text{ is internal,} \\ \sum_i d_i\mathbb{E}[z_{ik}]\sum_{j\neq i}\mathbb{E}\left[z_{jk}\right] d_j/4m, & r \text{ is terminal.} \end{cases}
\end{aligned} \tag{3.24}$$

Dynamic programming

Lazy evaluation of sufficient statistics A naïve implementation of the inference algorithm would require $O(K^2n^2)$ operations for the local and global updates, with K number of terminal-level nodes on a network of n vertices. On a sparse network, we could reduce n^2 to m, the number of edges, where we have $m \ll n^2$ and $O(m) = O(n)$. The idea is to describe update equations in terms of sufficient statistics, not edges and vertices.

We define the expected degree of vertex i to a cluster a,

$$d_{ia} \equiv \sum_j e_{ij}z_{ja} = \sum_{j:\, e_{ij}>0} e_{ij}z_{ja}, \tag{3.25}$$

the expected size of a cluster,

$$n_a \equiv \sum_j z_{ja} \tag{3.26}$$

and the expected volume,

$$v_a \equiv \sum_j d_j z_{ja}. \tag{3.27}$$

All the terms required for the update equations can be rewritten with respect to the above statistics; once we had the full calculation finished, we could easily update them by difference made by the changes made in latent variables. For instance, if we had a new $z_{ja}^{(t)}$ on t-th step of optimization from the old $z_{ja}^{(t-1)}$, we could reflect this change by setting

$$d_{ia} \leftarrow d_{ia} + e_{ij}(z_{ja}^{(t)} - z_{ja}^{(t-1)}).$$

Evaluation by recursion The required $O(K^2)$ scaling attributes to cluster-cluster dependency. However since it is represented by a tree structure, a majority of computation can be spared by divide-and-conquer and memoization. At an internal r any statistic s can be computed recursively, i.e.,

$$s_{i,r} = s_{i,L_r} + s_{i,R_r}$$

where s can be any of d_{ir}, n_r and v_r (Eq. 3.25, 3.26 and 3.27).

For instance, the mean-field local updates (Eq. 3.18 and 3.16) for a vertex i can be rewritten as

$$\begin{aligned}
\log Q(z_{ia} = 1|\cdot) &= \sum_{r \in [N]} \mathbb{E}_Q[\eta_r]^\top \sum_{b \in [K]:\, r = \mathrm{LCA(a,b)}} \mathbb{E}_Q[\mathbf{s}_{ib}] \\
&= \sum_{r \in [N]} \mathbb{E}_Q[\eta_r]^\top (d_{i,R_r}, n_{R_r})\, \mathbb{1}[a \in L_r] + \mathbb{E}_Q[\eta_r]^\top (d_{i,L_r}, n_{L_r})\, \mathbb{1}[a \in R_r]
\end{aligned}$$

with the equality up to some constant factor. Notice that all the terms, $\mathbb{E}_Q[\eta_r]$, d_{i,L_r}, d_{i,R_r}, n_{L_r} and n_{R_r}, are invariant to the choice of a, but only the indicator function varies depending on the location of a with respect to r. Therefore $\log Q(z_{ia} =$

1) could be evaluated for all $a \in [K]$ at once if sufficient statistics and global parameters were pre-calculated; i.e., we may visit the tree nodes in depth-first traversal to collect relevant terms. The same idea can be applied to other updates (Eq. 3.20).

Dynamic programming algorithm for local updates For a general description let

$$\mathbf{s}_{i,L_r} \equiv \sum_{a \in L_r} \mathbf{s}_{ia}, \quad \mathbf{s}_{i,R_r} \equiv \sum_{a \in R_r} \mathbf{s}_{ia}.$$

We also define the factors of left and right of r by

$$f_{\text{left}}(i,r) \equiv \begin{cases} \mathbb{E}_Q[\eta_r]^\top \mathbf{s}_{i,R_r} & \text{for the mean-field update,} \\ \log \mathbb{E}_Q\left[\exp\left(\eta_r^\top \mathbf{s}_{i,R_r}\right)\right] & \text{for the LCVI update} \end{cases}$$

and

$$f_{\text{right}}(i,r) \equiv \begin{cases} \mathbb{E}_Q[\eta_r]^\top \mathbf{s}_{i,L_r} & \text{for the mean-field update,} \\ \log \mathbb{E}_Q\left[\exp\left(\eta_r^\top \mathbf{s}_{i,L_r}\right)\right] & \text{for the LCVI update;} \end{cases}$$

and for the terminal node k we define

$$f_{\text{terminal}}(i,k) \equiv \begin{cases} \mathbb{E}_Q[\eta_k]^\top \mathbf{s}_{i,k} & \text{for the mean-field update,} \\ \log \mathbb{E}_Q\left[\exp\left(\eta_k^\top \mathbf{s}_{i,k}\right)\right] & \text{for the LCVI update.} \end{cases}$$

The overall algorithm proceeds in two stages: first we calculate partial factors f in depth-first order traversal; then summing over the factors along the paths from root and evaluate log probability of vertex i's assignment (Alg. 2).

Dynamic programming algorithm for global updates We can formulate a similar algorithm for the global updates. For the uncorrected model (Eq. 3.23), we

Alg 2 Recursive latent variable inference

repeat
  for all $i \in [n]$ **do**
    compute or update $\mathbf{s}_{ik}$ for all $k \in [K]$
    CALCPATH(root of $\mathcal{T}$)
    SUMPATH(root of $\mathcal{T}$, 0)
    $\mu_{ik} \propto \exp\{\nabla[k]\}$
  end for
until convergence

function CALCPATH(r)
  if r is leaf-level, cluster k **then**
    return $\mathbf{s}_{ik}$
  else
    $\mathbf{s}_{i,L_r} \leftarrow$ CALCPATH(left(r))
    $\mathbf{s}_{i,R_r} \leftarrow$ CALCPATH(right(r))
    $\delta_{\text{left}}[r] \leftarrow f_{\text{left}}(i, r)$
    $\delta_{\text{right}}[r] \leftarrow f_{\text{right}}(i, r)$
    return $(\mathbf{s}_{i,L_r} + \mathbf{s}_{i,R_r})$
  end if
end function

function SUMPATH(r, α)
  if r is terminal-level, cluster k **then**
    $\nabla[k] \leftarrow \alpha + f_{\text{terminal}}(i, k)$
  else
    SUMPATH(left(r), $\alpha + \delta_{\text{left}}$[r])
    SUMPATH(right(r), $\alpha + \delta_{\text{right}}$[r])
  end if
end function

rewrite the required statistics in terms expected cluster-wise degree (Eq. 3.25 and cluster size (Eq. 3.26). At the terminal level statistics, we have expected edges

$$\begin{aligned} \mathbb{E}[m_k] &= \frac{1}{2} \sum_{i \neq j} e_{ij} \mathbb{E}\left[z_{ik} z_{jk}\right] \\ &= \frac{1}{2} \sum_{i} \mathbb{E}[z_{ik}]\, d_{ik} \end{aligned}$$

and total counts

$$\begin{aligned}\mathbb{E}[t_k] &= \frac{1}{2}\sum_{i\neq j}\mathbb{E}[z_{ik}]\,\mathbb{E}\left[z_{jk}\right] \\ &= \sum_i \mathbb{E}[z_{ik}]\,(n_k - \mathbb{E}[z_{ik}])/2.\end{aligned}$$

At the internal level, we can recursively accumulate expected edges

$$\begin{aligned}\mathbb{E}[m_r] &= \sum_{a\in L_r, b\in R_r}\sum_{i,j} e_{ij}\mathbb{E}[z_{ia}]\,\mathbb{E}\left[z_{jb}\right] \\ &= \sum_i \sum_{a\in L_r}\mathbb{E}[z_{ia}]\sum_{b\in R_r}\sum_j e_{ij}\mathbb{E}\left[z_{jb}\right] \\ &= \sum_i \mathbb{E}[z_{i,L_r}]\,d_{i,R_r},\end{aligned}$$

and total counts

$$\begin{aligned}\mathbb{E}[t_r] &= \sum_{a\in L_r, b\in R_r}\sum_{i,j}\mathbb{E}[z_{ia}]\,\mathbb{E}\left[z_{jb}\right] \\ &= \sum_i \mathbb{E}[z_{i,L_r}]\,n_{R_r} \\ &= n_{L_r} n_{R_r}.\end{aligned}$$

The overall procedure is summarized in Alg. 3.

For the global update of degree-corrected model (Eq. 3.24), we only modify the updates of total counts, which can be defined with respect to volumes (Eq. 3.27). At the terminal level,

$$\begin{aligned}\mathbb{E}[t_k] &= \frac{1}{2}\sum_{i\neq j} d_i d_j \mathbb{E}[z_{ik}]\,\mathbb{E}\left[z_{jk}\right]/2m \\ &= \sum_i d_i \mathbb{E}[z_{ik}]\,(v_k - d_i\mathbb{E}[z_{ik}])/4m\end{aligned}$$

and at the internal level,

$$\begin{aligned}\mathbb{E}[t_r] &= \sum_{a\in L_r,\, b\in R_r}\sum_{ij} d_i d_j \mathbb{E}[z_{ia}]\,\mathbb{E}\left[z_{jb}\right] \\ &= \sum_i d_i \mathbb{E}[z_{i,L_r}]\,(v_{R_r} - d_i\mathbb{E}[z_{iR_r}])/2m.\end{aligned}$$

We can use a similar algorithm (Alg. 4).

Alg 3 Global updates for the stochastic block model

compute n_k for all $k \in [K]$
for all $i \in [n]$ **do**
  compute d_{ik} for all $k \in [K]$
  COLLECTSTAT(root of $\mathcal{T}$)
end for
for all r **do**
  update $\mathbb{E}[\eta_r]$ using $\mathbb{E}[m_r]$, $\mathbb{E}[t_r]$
end for

function COLLECTSTAT(r)
  if r is leaf-level, cluster k **then**
    $\mathbb{E}[m_k] \leftarrow \mathbb{E}[m_k] + 0.5\, d_{ik}\, \mathbb{E}[z_{ik}]$
    $\mathbb{E}[t_k] \leftarrow \mathbb{E}[t_k] + 0.5\, \mathbb{E}[z_{ik}]\ (n_k - \mathbb{E}[z_{ik}])$
    return (d_{ik}, z_{ik}, n_k)
  else
    $(d_{i,L_r}, z_{i,L_r}, n_{L_r}) \leftarrow$ COLLECTSTAT(left(r))
    $(d_{i,R_r}, z_{i,R_r}, n_{R_r}) \leftarrow$ COLLECTSTAT(right(r))
    $\mathbb{E}[m_r] \leftarrow \mathbb{E}[m_r] + z_{i,L_r} d_{i,R_r}$
    $\mathbb{E}[t_r] \leftarrow \mathbb{E}[t_r] + z_{i,L_r} n_{R_r}$
    return $(d_{i,L_r}, z_{i,L_r}, n_{L_r}) + (d_{i,R_r}, z_{i,R_r}, n_{R_r})$
  end if
end function

Other technical details

Initialization Variational inference algorithms not necessarily guarantee the convergence to global optima. To avoid bad local optima, we may restart the algorithm multiple times from random configuration. However, the model space grows super-exponentially and an algorithm may require exponentially many random restarts. Instead, we found that iterative bisections of network provide a good starting point. Since each bisection using the deterministic inference with 2 groups can be completed in $O(m)$, we can finish the whole initialization in $O(mK)$.

Alg 4 Global updates for the degree-corrected stochastic block model

compute n_k, v_k for all $k \in [K]$
for all $i \in [n]$ **do**
  compute d_{ik} for all $k \in [K]$
  COLLECTSTAT(root of $\mathcal{T}$)
end for
for all r **do**
  update $\mathbb{E}[\eta_r]$ using $\mathbb{E}[m_r]$, $\mathbb{E}[t_r]$
end for

function COLLECTSTAT(r)
  if r is leaf-level, cluster k **then**
    $\mathbb{E}[m_k] \leftarrow \mathbb{E}[m_k] + 0.5\, d_{ik}\, \mathbb{E}[z_{ik}]$
    $\mathbb{E}[t_k] \leftarrow \mathbb{E}[t_k] + d_i \mathbb{E}[z_{ik}]\ (v_k - d_i \mathbb{E}[z_{ik}])/4m$
    return (d_{ik}, z_{ik}, v_k)
  else
    $(d_{i,L_r}, z_{i,L_r}, v_{L_r}) \leftarrow$ COLLECTSTAT(left(r))
    $(d_{i,R_r}, z_{i,R_r}, v_{R_r}) \leftarrow$ COLLECTSTAT(right(r))
    $\mathbb{E}[m_r] \leftarrow \mathbb{E}[m_r] + z_{i,L_r} d_{i,R_r}$
    $\mathbb{E}[t_r] \leftarrow \mathbb{E}[t_r] + z_{i,L_r}\, (v_{R_r} - d_i z_{i,R_r})/2m$
    return $(d_{i,L_r}, z_{i,L_r}, v_{L_r}) + (d_{i,R_r}, z_{i,R_r}, v_{R_r})$
  end if
end function

Heuristics for speed up Although $O(mK)$ runtime is practical for small K, a network of 10,000 nodes and 100,000 edges could have K as large as 1000. Therefore, full computation of each variational update could make the overall algorithm scales essentially in $O(mn)$ (if $O(n) = O(K)$). We may reduce m as in the previous work [60] by stochastic variational inference. Here, we address different aspect, reducing K to some $k^* \ll K$.

Suppose we want to re-assign a vertex i by evaluating $\{\mu_{ik} : k \in [K]\}$. In assortative networks, vertices tend to form a group only with connected vertices. This allows us to carry out the computation of latent and global updates more efficiently, exploiting locality. Let U_i be a subset of leaf groups, to which the vertex

i is connected, i.e., $U_i = \{k \in [K] : d_{ik} > 0\}$. With a proper initial configuration (e.g., iterative bisections), we get $|U_i| \ll K$. Let $k_{\min} = \min U$ and $k_{\max} = \max U$. We now evaluate $\{\mu_{ik} : k \in [K_{\min}, K_{\max}]\}$ for the local updates.

Pruning unnecessarily branching subtrees Allowing sufficient depth of the tree model, we may generate an over-complicated model fitted to noisy observation. To reduce model complexity, we apply final pruning steps. At each subtree of the full model, we compared this subtree with the collapsed model under the single group. We determine whether to collapse or not via the Bayes factor, or log-ratio of the marginal likelihood [70]. This automatically determines the number of groups from the data.

3.4 Results

Performance evaluation

In this chapter we have established four variants of statistical inference methods. We may choose to fit regular stochastic block model (SB) or degree-corrected stochastic block model (DSB) using either locally collapsed variational inference (LCVI) or mean-field approximation (MF).

	locally collapsed	mean-field
stochastic block model	hSB-lcvi	hSB-mf
degree-corrected model	hDSB-lcvi	hDSB-mf

Table 3.1: Statistical inference methods.

Benchmark tests We generated a sparse network through the LFR benchmark [103]. Fitting on the training network data each method outputs a set of vertices, or group structure. Similarity to ground truth was quantified by normalized mutual information (NMI) [102]. The NMI scales in between 0 and 1, and higher value means higher similarity. The following methods were considered in experiments.

- k-Metis [90]: multi-level min cut algorithm. We fed a correct number of groups, k, or performed grid search over multiple k's and reported best performance.

- CNM [30]: Newman-Girvan modularity maximization algorithm. The CNM method resolves optimal group structure with respect to modularity score.

The benchmark program generated sparse network data with average degree 10 and maximum degree 100; we set minimum size of groups 10 and varied maximum size from 20 to 150 (the titles of columns in Fig. 3.2), and also varied number of vertices from $5,000$ to $10,000$.

The performance of hDSB dominates others (Fig. 3.2. The effect of different latent variable inference was not so significant, LCVI versus MF. For networks with balanced group structure, with the maximum group size ≤ 50, Metis algorithm performs nearly as well as hDSB. However, it requires the number of groups as a parameter, and real-world networks may contain heterogeneous group structures. The degree-correction provides more realistic group structure than regular stochastic block models; in fact, we found that it prevents from over-segmentation. Not surprisingly, since CNM algorithm relies on local greedy

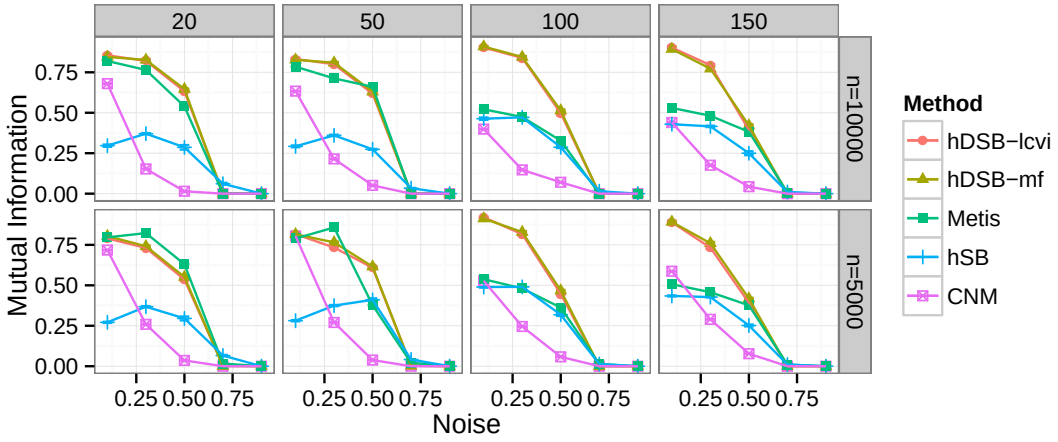

Figure 3.2: The benchmark result. *x-axis*: noise parameter (the μ parameter of LFR [103]); *y-axis*: normalized mutual information [102]; *titles on the columns*: the maximum size of a group; *titles on the rows*: total number of vertices. See the text for details.

steps, it was most sensitive to noise and resolution limit problem [45] (under-segmentation).

We performed benchmarks on more dense networks, e.g., average degree 20 or 50, but all the methods attained to similar level of performance. We omit the results here.

Link prediction Without gold-standards, performance may be assessed by link prediction. We used the same evaluation methods in the previous chapter (Chapter 2). We first remove links chosen uniformly from the observed network, and constructed a training data and a positive test set. We then uniformly and randomly selected the same number of pairs, which provides an unbiased negative test set [143, 144]. Given the training network, where known links were held out, we fit a model and predict links on the test set. Therefore each method generates scores s_{ij} for pairs ij in the test set.

Link prediction tasks do not require fixed and disjoint group structure, and permits comparison with mixed membership stochastic block models (mmSB) [3, 60] and full hierarchy of stochastic models, such as HAC-ML (Chapter 2). For mmSB we used a recent `C++` implementation provided by the authors [60], but the original batch inference algorithm did not scale to networks with more than 1000 vertices [3]. For hierarchical stochastic block models we considered three variants, HAC-ML [140], hSB [142] and hDSB.

CNM and k-Metis output disjoint groups / blocks. We estimated link prediction score s_{ij} of a pair ij by group-wise link frequency,

$$s_{ij} \approx \frac{\text{\# links observed between one end in group } a \text{ and the other in group } b}{\text{\# possible links between groups } a \text{ and } b}.$$

if i and j belong to groups a and b respectively. CNM method resolves number of groups automatically, but we performed exhaustive grid search on k of k-metis and reported best performance. For mmSB we compute score s_{ij} based on expected latent mixed membership assignment of vertex i to a group k, $\mathbb{E}[z_{ik}]$, and expected parameter of the block k, $\mathbb{E}[\beta_k]$, that is

$$s_{ij} = P(e_{ij} = 1|\mathsf{mmSB}) = \sum_{k=1} \mathbb{E}[z_{ik}]\, \mathbb{E}\left[z_{jk}\right]\, \mathbb{E}[\beta_k]$$

(see Eq.1 of Gopalan and Blei [61]). For the hierarchical models without degree correction, such as HAC-ML and hSB, we estimate score

$$s_{ij} = P(e_{ij} = 1|\mathsf{hSB}) = \sum_{a,b} \mathbb{E}[z_{ia}]\, \mathbb{E}\left[z_{jb}\right]\, \mathbb{E}\left[\theta_{\mathrm{LCA(a,b)}}\right].$$

For the degree corrected model, hDSB, we multiply additional factor $\hat{\rho}_{ij}$ estimated from training data and compute score

$$s_{ij} = P(e_{ij}|\mathsf{hDSB}) = \sum_{a,b} \mathbb{E}[z_{ia}]\, \mathbb{E}\left[z_{jb}\right]\, \mathbb{E}\left[\theta_{\mathrm{LCA(a,b)}}\right] \hat{\rho}_{ij}.$$

For hSB and hDSB we used LCVI, but results were largely invariant to the choice of inference method.

Link prediction experiments We constructed networks from all physical interactions available in BioGRID database (version 3.1.94) [174]. See Table. 3.2 for basic statistics. Although we performed link prediction experiments for all species, here we show only the results of *Drosophila melanogaster*, *Homo sapiens* and *Saccharomyces cerevisiae* (Fig. 3.3). We judged that networks of the other species were largely incomplete from low average degrees < 5.

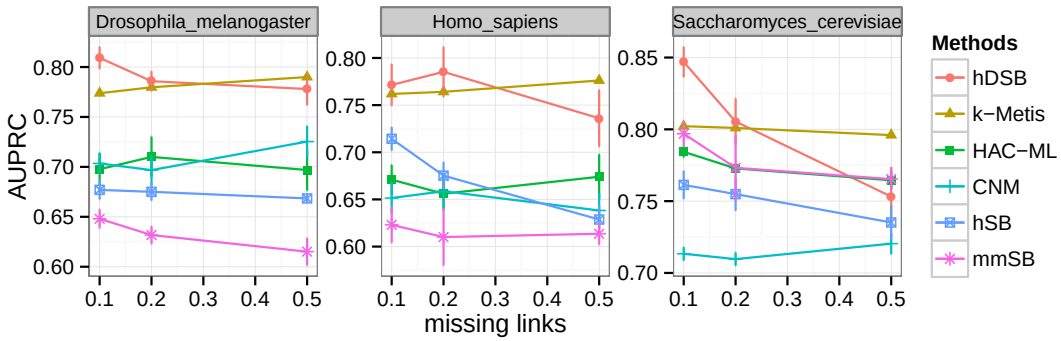

Figure 3.3: Link prediction performance on all physical interactions Different colors and shapes represent different methods (see the text). Vertical bars show magnitude of standard errors, estimated standard deviation $/\sqrt{n}$ with n experiments. Some dots graphically omit standard errors because of very small magnitude.

From the link prediction results, we gain multiple insights into biological networks (Fig. 3.3). We see hDSB and k-Metis (best one) algorithms show consistently best accuracy. Since the Metis algorithm searches for best possible k cuts, largely physical interaction networks divide into modules that are distinguishable by cuts. In the *S.cerevisiae* network, however, the degree-corrected stochastic model significantly better explain data. Considering this network was most

dense, substantial fraction of edges exists across modules, which may mislead k-metis algorithm.

Moreover, a fully branching tree model, obtained from the greedy HAC-ML algorithm, outperformed the hSB model found by more elaborate inference algorithm. We noticed the hSB, unlike hDSB, tends to clump small clusters into larger ones. Although the model itself has no limited, initialization made by the top-down iterative bisection was not quite suitable. However, in the degree corrected model iterative top-down bisection worked effectively.

We also notice that the mmSB models fit poorly to biological networks. Except for the *S. cerevisiae* network, the prediction accuracy was significantly lower than others and quite close to the random 0.5 level. Stochastic optimization algorithm used to fit the mmSB model could be one reason since the algorithm does not scan overall pairs, but samples a small fraction of pairs [60]. A full batch learning could improve performance [3]; however the batch learning scales in $O(K^2)$ and has little usage in practice. Even the stochastic learning was slowest of compared methods. Nevertheless, our results not necessarily rule out the possibility of mixed memberships in biological networks. We have seen values of mixed membership in a pattern detection problem [136].

3.5 Biological impact

We developed a class of statistical inference algorithms that can identify hierarchical modular structures in large-scale biological networks. A new algorithm based on the degree-corrected stochastic block model [89] best modeled hidden structure of networks. We also confirm that biological networks are generally

experiments	species	V	E	$\bar{d}$
all `physical` interactions	*Arabidopsis thaliana*	5783	13044	4.51
	Caenorhabditis elegans	2915	4670	3.20
	Drosophila melanogaster*	8001	34801	8.70
	Homo sapiens*	14948	83490	11.17
	Mus musculus	5129	9305	3.63
	Rattus norvegicus	1540	1794	2.33
	Saccharomyces cerevisiae*	6035	76674	25.41
	Schizosaccharomyces pombe	1770	3940	4.45

Table 3.2: Summary statistics of BioGRID physical interaction networks (3.1.94) [174] *Symbols*: V, number of vertices (genes/proteins); E, number of edges (unique interactions); $\bar{d}$, average degree. We restrict link prediction experiments on networks with average degree greater than or equal to 5 (marked by $\star$).

organized hierarchically and modularly [153, 154].

In order to understand the result more explicitly, we consult to the network conductance plot (NCP; Fig. 3.4). From the network conductance plot we can understand modular structure of a network [108]. For each cluster we can identify cut-edges crossing between the clusters and outside. The network conductance normalizes cut-edges by total possible number of cut-edges, so the conductance scales in between 0 and 1. The x-axis corresponds to size of clusters that increase logarithmically; the y-axis corresponds to network conductance of clusters. We also looked at NCP at the bottom level clusters (bottom; red) and higher level clusters (top; blue). Bottom-level clusters were identified by complete iterative bisection; top-level clusters were identified by follow-up full model learning, which tends to merge smaller pieces into larger clusters.

Modules are not well-defined by minimum cut, i.e., a small conductance value; instead, they are also highly connected to outside (Fig. 3.4). In a rough estimation only about 50% of edges are connected within clusters. Visual inspection

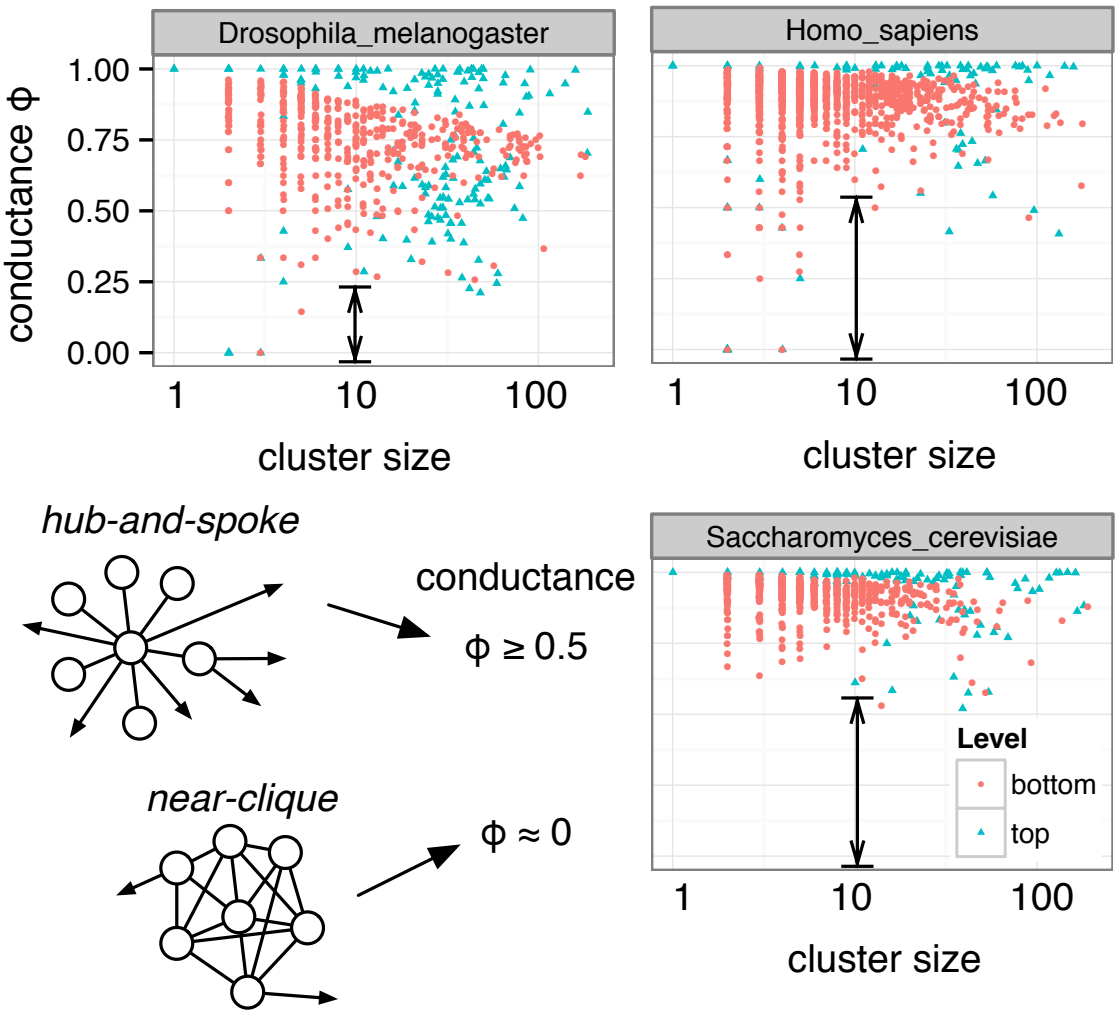

Figure 3.4: Network conductance plot. Both top and bottom-level clusters enrich a "hub-and-spoke" type of patterns, rather than "cliques."

suggests that most prevalent clusters were in a "hub-and-spoke" pattern. Within a module, one "local" hub gene holds actively engage with other members, but other members only sparsely connect to each other.

This finding suggests that within a functional module there is a small set of leading genes/proteins and others simply follow them. We may conjecture that the "local" hubs play a key regulatory function, but their function is confined

locally. Characterization of these local modulator using other information sources will illuminate detailed views on regulatory circuits.

3.6 Technical impact

Here we made technical improvements on general network clustering methods. A hierarchical structure lends an efficient dynamic programming algorithm that computes a pairwise group-group relations in a linear time scale. More importantly we wanted to design a tool that one can take advantage of the methods, just out of box. There is no tuning parameter in our network clustering algorithm. It automatically determines the size of model, and initialize the model with iterative bisection without having to run multiple times.

Underlying graphical models we considered are rather simple that the edges are independent given block membership of vertices. Even for the degree-corrected model we assume degree sequence is fixed beforehand, which retain conditional independence of the edges. The assumption was due to the structural / stochastic equivalence of a general framework (Def. 5 and Def. 6). However, note that our block-block structures are not independent. A possible interesting direction was pursued to redesign a stochastic block model in terms of triangular closure of edges, that we represent blocks more compactly, and a triangle could be indeed smallest unit of a block [72].

Chapter 4

Dynamic network modules by dynamic clustering and set matching algorithms

4.1 Introduction

We have sought to design reliable and fast algorithm that reveals hidden and hierarchical group structures in large-scale networks. Our agglomerative clustering method, HAC-ML, deterministically resolves multi-level structure of group of vertices, or communities. Improvement based upon statistical inference algorithm of deep but fixed tree models were also substantial in both benchmark and link prediction experiments.

However, large-scale compendia of interactions are primarily static lists that lack the dynamic aspects of living molecular systems. These interactions come primarily from high-throughput screens that may not be specific to a single temporal stage (such as affinity purification / mass spectrometry of yeast protein complexes obtained as an average over the cell cycle) or may involve an engineered system entirely removed from natural cellular dynamics (such as two-hybrid screens). Other interactions inferred from numerous bioinformatics methods, including cross-species inference, necessarily lack information about network dynamics.

Simulated dynamic networks The approach used here is to assume that interactions collected in a compendium represent a superposition of the possible interactions that could occur within a cell. From a different data source, we obtain a profile of the active network components. These data sets are joined in a probabilistic model, termed a dynamic hierarchical stochastic block model, to infer network evolution. Our application is to protein interaction networks, but the same techniques could be applied to other types of networks, or to a complex network of multiple interaction types. Dynamics of proteins are inferred from transcript presence or absence in mRNA profiling studies, an admittedly inaccurate proxy for protein levels but nevertheless the primary type of dynamic data readily available for cellular systems.

Dynamic clustering methods Here we propose dynamic extension of previously considered static methods, agglomerative clustering (Chapter. 2) and statistical inference algorithm (Chapter. 3). First extension made on to HAC-ML is to kernelize merging (Eq. 2.5) or collapsing scores (Eq. 2.7) of multiple snapshot networks. On each snapshot we resolve top and bottom groups assuming temporal smoothness. We term the new agglomerative clustering method dynamic hierarchical agglomerative clustering, or DHAC [141]. However different snapshot modules may have inconsistent indexes, so called an identifiability problem. To identify a chain of related modules we propose dynamic set matching algorithm that can help visualize dynamics. We deigned a new set matching algorithm by Expectation Maximization (EM), termed Match'EM [141]. Second extension builds on statistical inference method of a fixed tree model. We take into accounts of smoothness of neighboring snapshots, explicitly incorporating distance

between snapshots to the static objective function (Eq. 3.15). We estimate dynamic models time-constrained mean-field approximation. We term this method dynamic hierarchical model, or DyHM [142]. See summary table (Tab.4.1).

We applied our methods to reveal dynamics of yeast metabolic cycle (YMC) and *Arabidopsis* root developmental process. YMC transcriptional profiling reveals three dominant metabolic states: reductive building (RB, 977 genes); reductive charging (RC, 1510 genes); and oxidative (OX, 1023 genes) [186]. Almost a half of total genes oscillate along this cycle, indicating that a broad swath of processes are involved but making it difficult to extract specific dynamical modules from expression data alone. We used DHAC and Match'EM.

Another application is to dynamic evolution of protein networks required for root development in *Arabidopsis*, based on a classic data set generated by Benfey's laboratory [20]. The physical interactions used in this study are obtained from work by Geisler-Lee, Provart and coworkers [52] and available in The Arabidopsis Information Resource (TAIR) `ftp://ftp.arabidopsis.org/home/tair/Proteins/`. We fitted spatiotemporally evolving hierarchical models by dynamic variational inference algorithm.

	base method	basic idea	application
DHAC and Match'EM	HAC-ML (Chapter. 2)	kernelization and set matching	yeast metabolic cycle
DyHM	mean-field approximation (Chapter. 3)	additional objective function for temporal smoothness	*Arabidopsis* root development

Table 4.1: Summary of extended methods for dynamic network data

4.2 Dynamic hierarchical agglomerative clustering

Suppose we observe T snapshots of time-ordered networks, $\{G^{(t)} : t = 1, \ldots, T\}$. Each single network $G^{(t)} = (V^{(t)}, E^{(t)})$ consists of undirected and unweighted binary edges $E^{(t)}$ and vertices $V^{(t)}$. Vertices correspond to proteins, and edges represent possible protein-protein physical interactions (PPI). For an arbitrary pair, $t \neq t'$, $G^{(t)}$ and $G^{(t')}$ can have different vertices and edges.

The goal is to infer a corresponding sequence of time-evolving stochastic block models, $\{\mathcal{M}^{(t)} : t = 1, \ldots, T\}$, where each $\mathcal{M}^{(t)}$ is a good network-generative model for $G^{(t)}$. Many methods maximize the model for each snapshot independently, obtaining $\hat{\mathcal{M}}(t)$ as $\arg\max_{\mathcal{M}} P(\mathcal{M}|G^{(t)})$, and then attempt to stitch together the results. Here we show that introducing explicit coupling between time points improves dynamic network clustering.

Agglomerative clustering algorithm

We take the same strategy as HAC: build a guide tree, then collapse the tree. The question remains to how to couple neighboring snapshots in building maximum likelihood guide tree and collapsing.

Kernel-reweighted scores Kernelization of the scores λ and ϕ couples nearby snapshots, also providing noise reduction and smoothing. Merging and collapsing scores were kernelized using Gaussian Radial Basis functions with width parameter τ, $w(\Delta t, \tau) \propto \exp\{-|\Delta t|/\tau\}$, where for simplicity Δt is the difference in snapshot indices. The kernelized merging score $\lambda^K(t)$ and collapsing score $\phi^K(t)$

for the t^{th} snapshot (K denotes kernelized) are

$$\lambda_{12}^{K}(t;\tau) = \sum_{s=1}^{T} w(t-s,\tau)\,\lambda_{12}^{S}(s) \tag{4.1}$$

$$\phi_{12}^{K}(t;\tau) = \sum_{s=1}^{T} w(t-s,\tau)\,\phi_{12}^{S}(s). \tag{4.2}$$

Although the same clustering is used across all T time points, the scores will differ when proteins (or interactions) are present in one time point and absent in another. Kernels are normalized as $\sum_{s=1}^{T} w(t-s,\tau) = 1$. As $\tau \to 0$, $\lambda^K \to \lambda^S$ and $\phi^K \to \phi^S$. Collapsing is then performed as for single snapshots, stopping at the maximum of the bottom-up sum, termed $\phi^K(t;\tau) = \sum_{(i,j)\in\text{collapsed}} \phi_{ij}^K(t;\tau)$. The overall algorithm is summarized in Alg. 5.

In the DHAC-local method, the bandwidth parameter τ for snapshot t was selected from a grid-search over τ values $0.5, 1.0, 1.5, \ldots, 3.5$ to maximize $\phi^K(t;\tau)$, with smaller τ favored when the network changes quickly. For the network considered here, $\tau \approx 1$ to 2 depending on t. Alternatively, a constant value of τ may be used for all values of t, which we termed DHAC-constant. We set $\tau = 1$ for DHAC-constant, although in principle τ could be optimized by maximizing $\sum_t \phi^K(t;\tau)$. In practice, results were very robust to the value of τ, and the performance of DHAC-local was nearly identical to DHAC-constant with $\tau = 1$ (see Results).

Evaluation of the dynamic method

At each time point, we randomly select pairs of vertices (u, v), some connected at time t with $e_{uv}(t) = 1$, and others unconnected with $e_{uv}(t) = 0$, the relative fraction of connected pairs (edges) and unconnected pairs (holes) matching the

Alg 5 DHAC

for $t \leftarrow 1 \ldots T$ **do**
  Set each vertex to be a single cluster
  Let $\phi_{\text{cum}} \leftarrow 0$ be cumulative model comparison score (Eq. 4.2)
  Compute merging scores (Eq. 4.1) of pairs having an edge or one or more shared neighbors
  repeat
    Pick a pair i, j of maximum $\lambda_{ij}^{K}(t;\tau)$
    Update scores of affected pairs after merging i, j
    Merge i, j to i'
    Re-compute scores for merging $i', j \in \{j : e_{i'j} > 0 \vee \sum_k e_{i'k} e_{kj} > 0\}$.
    Update $\phi_{\text{cum}}(t;\tau) \leftarrow \phi_{\text{cum}} + \phi_{ij}^{K}(t;\tau)$
  until no more pairs to merge
  Output group structure $\mathcal{M}(t;\tau)$ at which $\phi_{\text{cum}}(t;\tau)$ was maximum
end for

network as a whole. These pairs are then a test set, and the remaining edges serve as the training set. After clustering based on the training set, vertex u will be assigned to some group i, and vertex v will be assigned to group j. The maximum likelihood probability of the (u, v) edge, denoted $\hat{e}_{uv}^{(t)}$, is then $\hat{e}_{ij}^{(t)} = e_{ij}^{(t)} / (e_{ij}^{(t)} + h_{ij}^{(t)})$.

Competing methods We compared the following algorithms: DHAC-constant, dynamic clustering with a constant fixed bandwidth ($\tau = 1$ for the link prediction experiments); DHAC-local, bandwidths adaptively optimized for each snapshots ($\tau = 0.5, 1.0, 1.5, \ldots, 3$); HAC, DHAC with bandwidth $\tau = 0$; and CNM, fast modularity optimization [30].

Drosophila networks As a proof of concept we first tested our algorithm on a dynamic network for *Drosophila* development, for which a gene expression time course is available [6]. Rather than analyzing the expression data directly, we

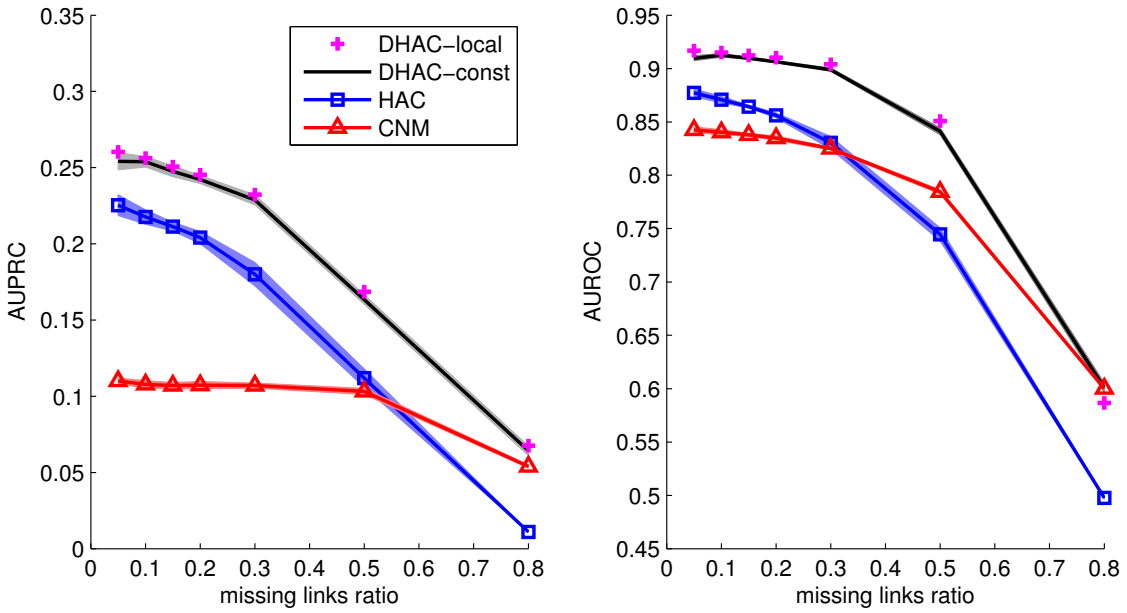

Figure 4.1: Link prediction results for *Drosophila* networks. *Left:* cumulative AUPRC scores for different methods (y-axis) along different missing link ratios (x-axis); *Right:* AUROC scores for different methods (y-axis) along different missing link ratios (x-axis). *Points and lines:* averge time-cumulative performance; *shaded area:* 1-standard error. See Methods for details.

relied on previous analysis using KELLER to identify time-varying regulatory interactions between genes, yielding a network with 66 time points and 588 gene vertices [172]. Thus, genes u and v are connected at time $t \in 1 \ldots 66$ according to these previous results, defining a sparse time-varying network with mean vertex degree ≈ 6.5. Since gene interactions were generated with time smoothing, DHAC-constant and DHAC-local are expected to outperform static methods.

In extensively cross-validated link prediction performance, DHAC-constant and DHAC-local are seen to be far superior to the next-best method, HAC, which in turn dominates CNM until at least 30% of the true edges are removed (Fig. 4.1). To perform these studies, from 5% to 80% of the known edges were removed; results were averaged over the 66 time points; and the entire procedure was re-

peated ≥ 10 times.

The similar results of DHAC-constant and DHAC-local point to robust behavior with respect to the kernelization parameter τ. The improved performance of CNM relative to HAC at a high frequency of missing links may be due to the tendency of CNM to generate large clusters and to lose resolution. The resolution limit is usually a drawback, but here is beneficial for link prediction in a sparsified network. Even in this limit, however, DHAC remains superior by drawing information from adjacent time points.

4.3 Dynamic hierarchical model

We introduce an extension in which group-group interactions are constant over space and time, but group membership can vary dynamically. But we additionally believe that an abrupt change between $Q^{(t)}$ and $Q^{(t')}$ is rare when times t and t' are adjacent. Note also that the index t is more general than a sequential time index, and we think more generally of the set of snapshots t' that are neighbors of snapshot t. So, we consider this divergence as well in the following objective function:

$$\mathcal{F} \;=\; D_{\mathrm{KL}}\Big(P^{(t)}\Big\|Q^{(t)}\Big) + \lambda \sum_{s\in N(t)} D_{\mathrm{KL}}\Big(Q^{(t)}\Big\|Q^{(s)}\Big) \tag{4.3}$$

The first term provides a conventional mean-field approximation between a true model distribution $P^{(t)}$ and the surrogate factorized $Q^{(t)}$, and the second handles our belief in spatiotemporal smoothness. In other words, we want to find $Q^{(t)}$ as close as possible to $P^{(t)}$, but not very apart from the neighboring snapshots $s \in N(t)$. We term our novel approach a Dynamic Hierarchical Model (DyHM).

We note there is in fact only one adjustable parameter, λ, which controls the spatiotemporal smoothness. Setting $\lambda = 0$ is equivalent to treating the snapshots as if they were independent, and large λ gives static group membership. The remaining parameters are all optimized as part of the model and are not subject to tuning. Furthermore, the model likelihood can be used as a guide for selecting λ itself, leading to a model with no adjustable parameters, other than the depth selected for the hierarchical tree.

Latent variable update The role of the adjustable parameter λ can be seen explicitly in the latent variable update. For algebraic convenience, we account for time-dependency among active genes by introducing auxiliary variables: let $m_i(t) = 1$ indicate that gene i is active at time t, and $m_i(t) = 0$ if inactive.

Suppose we minimize the objective (Eq. 4.3) readjusting the latent assignment of a vertex i with respect to snapshot tree $\mathcal{T}^{(t)} = (N^{(t)}, \mathcal{C}^{(t)}, \Theta^{(t)})$ at certain time t. Overall derivation is identical to the static mean-field update (Eq. 3.18); therefore we omit details because it is tedious. We have the following local update equation for each vertex i and bottom-level a:

$$Q(z_{ia}^{(t)} = 1|\cdot) \propto \exp(H) \tag{4.4}$$

where

$$H \equiv \frac{\sum_{r\in[N]} \phi_r + \lambda \sum_{t'} m_i(t')m_i(t) \log Q(z_{ik}^{(t')}|\cdot)}{1 + \lambda \sum_{t'} m_i(t')m_i(t)}$$

and

$$\phi_r \equiv \mathbb{E}_Q[\eta_r]^\top \sum_{b\in[K]:\, r=\mathrm{LCA(a,b)}} \mathbb{E}_Q[\mathbf{s}_{ib}]\,.$$

From the above (Eq. 4.4), we can consider two extreme cases:

$$\begin{aligned}\lim_{\lambda\to 0} H &= \sum_{r\in[N]} \phi_r, \\ \lim_{\lambda\to\infty} H &= \frac{\sum_{t'} m_i(t')m_i(t)\log Q(z_{ia}^{(t')}=1|\cdot)}{\sum_{t'} m_i(t')m_i(t)}.\end{aligned}$$

The first assumes independence between time points, while the latter approximates the current position by the geometric mean of adjacent ones.

Simulation study

Dynamic synthetic data. The dynamic data was generated by assigning 30 total vertices initially to 5 groups. A snapshot of a set of edges was then generated by adding within-group edges to the snapshot with probability $\mathsf{P}_{\mathsf{within}}$, and adding between-group edges with probability $\mathsf{P}_{\mathsf{between}}$. After each snapshot, the edges are erased, each vertex switches to a different group at random with probability $\mathsf{P}_{\mathsf{switch}}$, and the process continues. This process permits the number of vertices in each group to change with time. The known group assignments provide a gold standard of known positives to assess the inferred co-membership probabilities.

Results from DYHM using a depth-3 hierarchy (8 groups) at various values of λ, including extreme values corresponding to independent and superimposed snapshots, were compared with co-membership inferred by the hypergeometric method [58]. For each snapshot we generated a PR curve and a corresponding F_1 score (the maximum harmonic mean of precision and recall along the curve).

Co-membership scores The co-membership probability of two different vertices i and j is computed from the posterior probability $Q(z_{ik})$ trained. The prob-

ability of these vertices being co-clustered is

$$p(\exists k, z_{ik} = 1 \wedge z_{jk} = 1) \stackrel{def}{=} \sum_k Q(z_{ik} = 1|\cdot)Q(z_{jk} = 1|\cdot)$$

where we do not consider the special case $i = j$.

Performance on dynamic networks. On relatively easy data sets ($\mathsf{P}_{\mathsf{within}} > 0.6$ and $\mathsf{P}_{\mathsf{between}} < 0.3$), all models work well. On harder simulation tests, however, DYHM gave superior performance. An example is $\mathsf{P}_{\mathsf{within}} = 0.5$, $\mathsf{P}_{\mathsf{between}} = 0.3$, and $\mathsf{P}_{\mathsf{switch}} = 0.05$ (Fig. 4.2). The value of λ selected by penalized likelihood (which requires no knowledge of the true group assignments) also gives the best performance in predicting time-dependent co-membership, $F_1 \approx 0.9$ corresponding to roughly 90% precision and recall. It performs better than independent analysis of each static snapshot, corresponding to $\lambda = 0$, with $F_1 \approx 0.8$. We note that the $\lambda = 0$ version of DYHM itself outperforms the hypergeometric predictor, which gives $F_1 \approx 0.7$.

We further tested the ability of λ to track networks with increasingly labile group membership, ramping $\mathsf{P}_{\mathsf{switch}}$ through values 0.01, 0.05, 0.2, 0.3, and 0.5, on non-trivially simulated network data with $\mathsf{P}_{\mathsf{within}}$ and $\mathsf{P}_{\mathsf{between}}$ respectively fixed at 0.5 and 0.3.

4.4 Dynamic set matching by expectation maximization

DHAC-constant and DHAC-local output T models, $\{M_1, \ldots, M_T\}$, and many groups will change slowly between time points. The total number of groups may differ between time points, however, and even if the number of groups and the group membership are nearly identical, group order may be permuted across

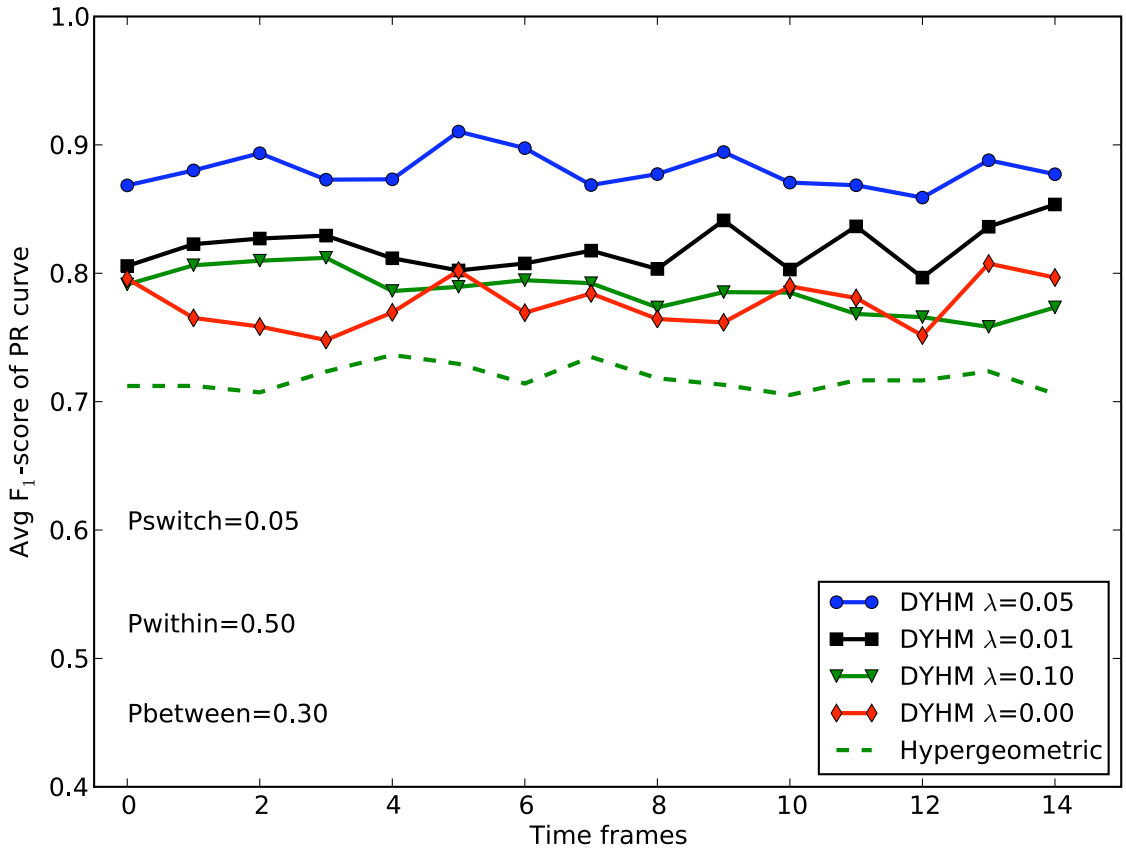

Figure 4.2: Comparison on dynamic synthetic networks. From top to bottom, lines denote correspond to F_1 scores over time frames. *Blue circle*: DYHM with $\lambda = 0.05$. *Black square*: DYHM with $\lambda = 0.01$. *Green triangle*: DYHM with $\lambda = 0.1$. *Red diamond*: DYHM with $\lambda = 0$. *Dashed green*: Hypergeometric method [58] applied separately to each each time frame.

time points. Matching similar groups across time points remains a general problem for dynamic networks. For a single pair of models, reasonable yet *ad hoc* procedures are to match groups based on shared members, Jaccard correlation of shared neighbors, or maximum weighted matching of shared neighbors or other pairwise scores [18]. Here we extend these ideas to multi-partite matching based on a novel probabilistic model that introduces some rigor to the time course matching problem.

The goal is to find most probable mapping of cluster i at time t to a globally consistent index k. Let $z_{ik}^{(t)} = 1$ if cluster i of snapshot t is assigned to k, and 0 otherwise, with normalization $\sum_k z_{ik}^{(t)} = 1$. Conversely, the sum over local clusters, $\sum_i z_{ik}^{(t)}$, is not fixed because the global cluster may be absent at time t (sum = 0) or it may be broken into multiple smaller clusters (sum > 1).

Each cluster i contains original network vertices $\{u\} \subseteq V$, and $n_{ij}^{(t)}$ counts the number of shared members between group i at time t and group j at time $t+1$. The probability that a vertex makes a transition from global state k to state k' between two snapshots is $\psi_{kk'}$, with normalization $\sum_{k'} \psi_{kk'} = 1$. For simplicity, $\psi_{kk'}$ is independent of t. When groups do not change over time, $\psi_{kk'} = \delta_{kk'}$, 1 if $k = k'$ else 0. Similarly, the time-independent parameter ν_{uk} is the probability that vertex u is in global group k, with $\sum_k \nu_{uk} = 1$.

The matching probability under consistent indexing is

$$P(\{M_t\}, \{z_{ij}^{(t)}\}|\nu, \psi) = \prod_{k=1}^{K} \prod_{t=1}^{T} \prod_{i \in S_t} \prod_{u \in C_i} \nu_{uk}^{z_{ik}^{(t)}} \times \prod_{k=1}^{K} \prod_{k'=1}^{K} \prod_{t=1}^{T-1} \prod_{i \in S_t} \prod_{j \in S_{t+1}} \psi_{kk'}^{n_{ij}^{(t)} z_{ik}^{(t)} z_{jk'}^{(t+1)}} \tag{4.5}$$

where S_t denotes the set of clusters at snapshot t and C_i the set of vertices in one of these clusters.

We solved the *maximum a posteriori* (MAP) inference problem using Expectation-Maximization (EM). The M-step updates are

$$\nu_{uk} \propto \sum_{t=1}^{T} \sum_{i \in S_t} z_{ik}^{(t)} I\{u \in C_i\}, \tag{4.6}$$

$$\psi_{kk'} \propto \sum_{t=1}^{T-1} \sum_{i \in S_t} \sum_{j \in S_{t+1}} n_{ij} z_{ik}^{(t)} z_{jk'}^{(t+1)}. \tag{4.7}$$

The E-step for $z_{ik}(t)$ is more complicated. If the state at time t is represented as

the assignment matrix $\{z_{ik}(t)\}$, then the probability structure is a hidden Markov model (HMM). This state space is large, however, on the order of $K^K \sim K!$, because each of the approximately K clusters at time t may be assigned to one of K global clusters, and the transition matrix is of order K^{2K}. Instead, we simplify the state space by considering each $z_{ik}(t)$ independently and introducing additional couplings that create loops in the corresponding graphical model, no longer permitting a dynamic programming solution. When groups are stable over time, however, the topology is close to a tree structure and belief propagation (BP) works well [199].

For max-product BP algorithm we reformulate the above Markov Random Field, or joint probability (Eq. 4.5), constructing a factor graph consisting of factors (hyper-edges) and variables (latent variables). Latent variables $z_i^{(t)}$ take on values from $1, \ldots, K$, or succinctly $[K]$. In other words, $z_i^{(t)}$ provides the index k of the global cluster for which $z_{ik}^{(t)} = 1$. Parameters $\{\nu\}$ are used to represent singleton factors and $\{\psi\}$ pairwise factors. A certain latent variable $z_i^{(t)}$ depends on neighboring pairwise factors $N(i, t-1)$ from the previous snapshot and $N(i, t+1)$ from the subsequent snapshot. MAP inference is carried out by sending messages from i to j via pairwise factor e. The update equations of the message $m_{i \to e}$ from variable i at time t to factor e and then the message $m_{e \to j}$ from e to variable j at time $t+1$ is

$$m_{i \to e}(k) \propto \prod_{u \in \mathbf{C}_i} \nu_{uk} \prod_{f \in N(i,t-1) \cup N(i,t+1) \setminus \{e\}} m_{f \to i}(k) \tag{4.8}$$

$$m_{e \to j}(k) \propto \max \left\{ l \in [K] : \psi_{lk}^{n_{ij}^{(t)}} m_{i \to e}(l) \right\}. \tag{4.9}$$

For variable j at time $t-1$, the message $m_{e\to j}$ is

$$m_{e\to j}(k) \propto \max\left\{ l \in [K] : \psi_{kl}^{n_{ji}^{(t-1)}} m_{i\to e}(l) \right\}. \tag{4.10}$$

The belief b_i of a certain variable i at snapshot t is the product of incoming messages,

$$b_i(k) \propto \prod_{e\in N(i,t-1)\cup N(i,t+1)} m_{e\to i}(k), \tag{4.11}$$

normalized as $\sum_k b_i(k) = 1$. To prevent the MLEs and BP steps from overshooting, parameters and messages were updated as 1/10 of the full change, with updates to messages performed on a logarithmic scale. We call this EM method MATCH-EM (Alg. 6).

Alg 6 MATCH-EM

```
Initial greedy matching
Initialize ν and ψ
repeat
    repeat
        while forward and backward visit of factors do
            Calibrate messages i to j (Eq. 4.8, 4.9, 4.10)
        end while
        for each variable i do
            Update belief b_i (Eq. 4.11)
        end for
    until convergence of BP
    Update latent variables z_ik = 1 with k = argmax_l b_i(l) and z_ik' = 0 for other
k' ≠ k.
    Update ν̂, ψ̂ by MLE (Eq. 4.6, 4.7)
until convergence of EM
```

4.5 Yeast Metabolic Cycle (YMC) dynamics

Preparation of dynamic networks Dynamic biological networks were obtained by combining experimental gene expression time series data with static protein interaction networks to project out the consistent edges, both active (two interacting proteins are expressed) and inactive (neither protein is expressed). This method assumes that presence of a protein is related to transcriptional abundance of the corresponding transcript at a nearby time, with possible delays due to translation and protein lifetimes.

Time-series measurements of the expression levels of N genes across T time points generate a $N \times T$ matrix X. Each element X_{ut} corresponds to the expression of gene u at snapshot t. The matrix X is assumed to be pre-processed and normalized, here performed with `gcrma` quantile-normalization [195]. Next it is row-standardized to have zero mean, $\sum_t X_{ut} = 0$, and equal variance, $\sum_t X_{ut}^2 = T - 1$, for each gene.

The dynamics of the network were then inferred from X, under the assumption that proteins in a complex have correlated gene expression profiles [81]. To account for transient complexes and cases where delays due to translation and protein lifetime are important, correlations were averaged over a bandwidth τ,

$$\tilde{X}_{uv}(t) = \sum_{s=1}^{T} w(t-s,\tau) X_{us} X_{vs}$$

with the Gaussian kernel function $w(\Delta t, \tau) \propto \exp(-|\Delta t|/\tau)$ and normalized to 1. While this bandwidth τ has a similar role to the bandwidth for likelihood kernelization, it was not optimized but rather set to 1.5. Results were quantitatively similar for τ from 1.2 to 2. Smaller values of τ result in stricter co-expression requirements and result in a sparser network.

Each edge is then declared present or absent based on the value of $\tilde{X}_{uv}(t)$: for each snapshot $t = 1, \ldots, T$, a dynamic edge $e_{uv}(t) = 1$ if and only if $\tilde{X}_{uv}(t) > 0$ and $e_{uv} = 1$ in static network. This procedure retains edges at time t where both proteins are present ($X_{us}, X_{vs} > 0$) or both absent ($X_{us}, X_{vs} < 0$) for times s close to time t. We found that using the negative evidence improved the prediction of protein complexes, and that the transcriptional data could then be used to identify which complexes or subunits were present or absent at each time point. Results were stable for less stringent thresholds, $\tilde{X}_{uv}(t) > -0.5$. While this method is appropriate for periodic processes, other methods for extracting time-dependent interactions may be more appropriate for more general processes.

Prior to clustering, the network used for link prediction was made less sparse by applying an iterative degree cutoff (≥ 3). Combining with the 36 time-varying snapshots, 3 complete cycles of 12 snapshots each, reduced the size of the network from 1380 proteins per snapshot to 480 ± 14 and increased the mean vertex degree from 1.8 to 6.6. Networks were clustered by DHAC-local. Clusters were matched across time points using MATCH-EM to yield 31 complexes with a total of 613 proteins.

We checked robustness using a bootstrapping procedure in which a fraction α of edges are randomly rewired according to the degree-consistent configurational model [88]. We used $\alpha = 0.01$ and performed 500 bootstraps, with about 80% co-membership conserved across bootstraps at each snapshot.

Macro-view of YMC complexes We recovered 31 dynamic complexes with at least 3 proteins and bootstrap co-membership at least 80% (Fig. 4.3). Many of the complexes have cluster-specific gene ontology (GO) keywords with p-value

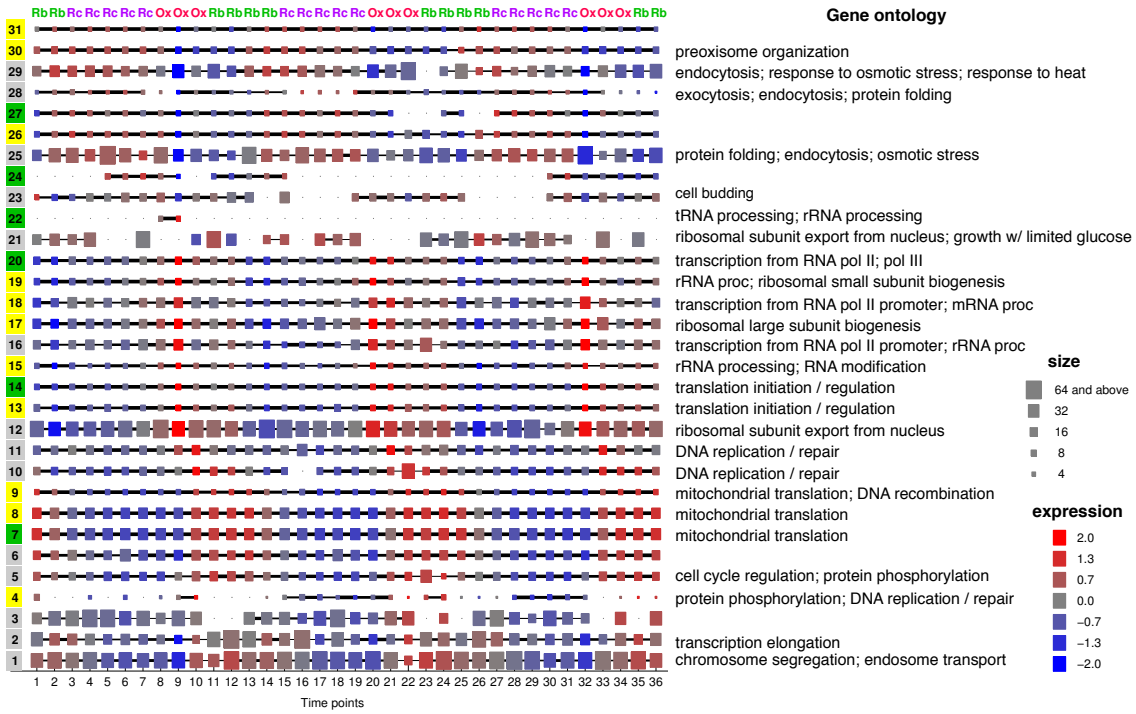

Figure 4.3: Dynamic network clustering reveals a detailed global view of periodic protein complexes during the yeast metabolic cycle. Squared nodes represent clusters matched across time points, showing only clusters having at least 3 genes/proteins. *Cluster order:* clusters are organized by peak activity in RB phase (#1 to #10), OX phase (#11 to #20), and RC phase (#23 to #31). *Node size:* number of genes/proteins contained in this cluster. *Node color:* average standardized gene expression level at time t. *Edge width:* Jaccard coefficient (or coherence) between clusters of adjacent snapshots. *Gene Ontology:* cluster-specific GO keywords were identified by hypergeometric tests.

≤ 0.05. Organizing clusters by average gene expression at each time point separates those that are active in each phase. RB clusters, #1 to #10, are related to cell cycle checkpoints and mitochondrial translation. OX clusters, #11 to #20, include ribosome metabolism, DNA replication/repair, and translation. RC clusters, #23 to #31, include stress response and transport. Most of the complexes can be matched across the entire time course, but some disappear then reappear. An example is complex #4, annotated for DNA repair, which is most active at the

end of each 12-point cycle. This behavior required the MATCH-EM algorithm for globally consistent clusters, and would have been impossible to resolve given matching to nearest neighbors alone.

We ascertained whether the complexes predicted by our methods correspond to known complexes obtained from manual curation, CYC2008, or from high throughput experiments, YHTP2008 [150]. The 408 manual and 400 high throughput complexes were filtered to retain the periodic proteins from YMC data, and then the catalog complex with the best Jaccard correlation was identified for each predicted complex. Of the 31 predicted complexes, 14 are poorly represented in the catalogs (Jaccard correlation $< 20\%$), 11 are only moderately similar (correlation $\geq 20\%$ and $< 80\%$), and 6 have a good match (correlation $\geq 80\%$). The predicted complexes with poor overlap often recombine subunits from multiple catalog complexes (see #16 below).

To test the effects of the filtering, we also performed clustering using all 63,410 BioGrid interactions and including all genes with YMC data, periodic or non-periodic, yielding a network of 54,758 interactions among 4987 proteins. Clustering this network and retaining complexes with at least 3 proteins and edge density > 0.1 yields 20 to 40 clusters at each snapshot with 900 ± 100 proteins included. Most clusters in the unfiltered network contain a high-degree core from the filtered network. Occasionally multiple cores are combined by low-degree connections, making the cluster count smaller than in the filtered network. The overlap with protein complex catalogs is similar to the unfiltered network.

Micro-views of YMC dynamics The protein complex dynamics provide a rich view of YMC providing new biological insight, as demonstrated by in depth anal-

ysis of clusters #7, the mitochondrial ribosome, and cluster #16, the nuclear pore.

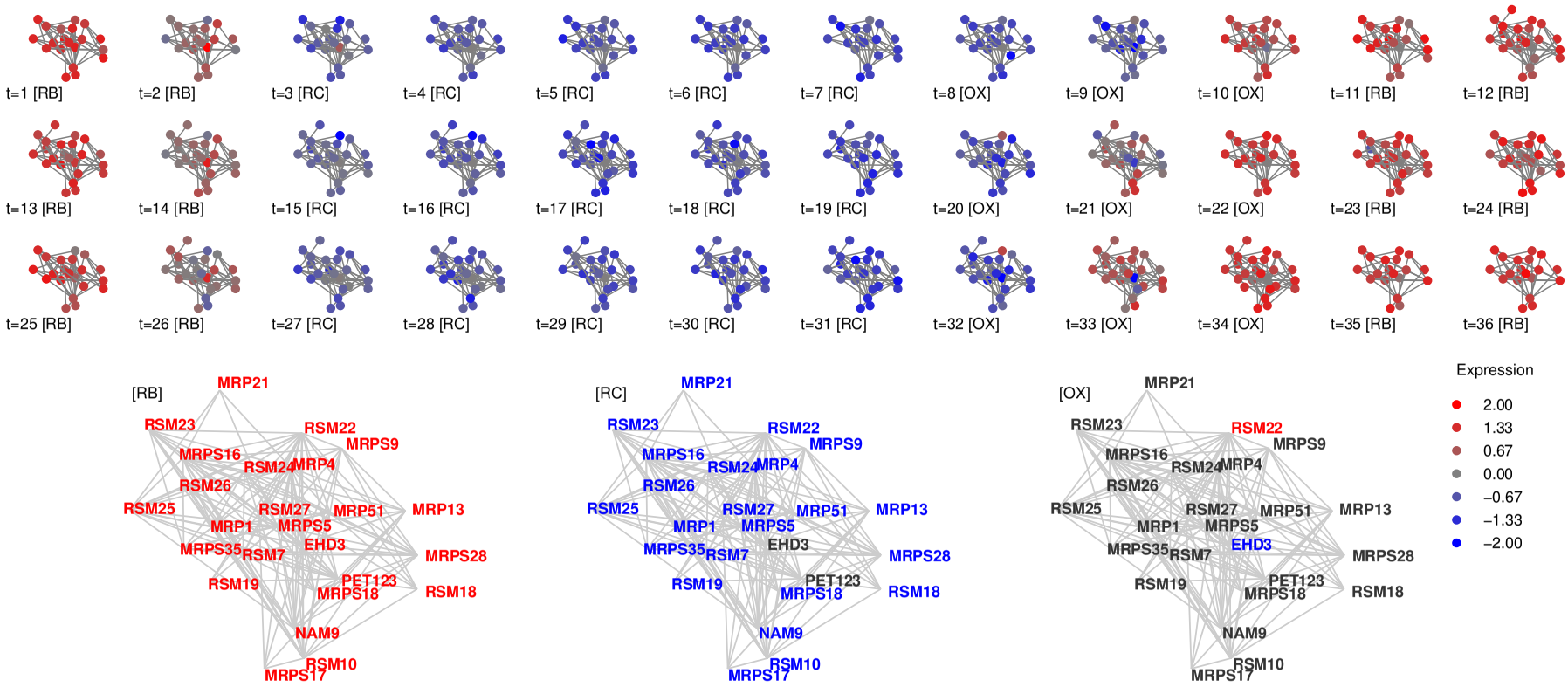

Figure 4.4: Cluster #7, mitochondrial ribosome. *Top:* cluster members for the 32 gene expression snapshots. *Bottom:* Average expression for the 3 YMC phases. *Node color:* standardized gene expression level. Gene names were colored red or blue if expression values are above 0.5 or -0.5 respectively.

Mitochondrial ribosome complex (#7) The mitochondrial ribosome is generally assumed to be RB-specific, with transcription switched on briefly at the transition from OX to RB (Fig. 4.4). This complex contains primarily RSMs (ribosomal small subunit of mitochondria) and MRPs (mitochondrial ribosomal protein), known components of the mitochondrial ribosome [164].

Underneath this general pattern, however, RSM22 shows systematic expression ahead of other components. At time points t=9, t=20, and t=32, RSM22 is active while other proteins are not transcribed. RSM22 is a nuclear-encoded putative S-adenosylmethionine (SAM) methyltransferase [147], and methylation of the $3'$ end of the rRNA of the small mitochondrial subunit is required for the assembly and stability of the mitochondrial ribosome [121]. Deleting RSM22 yields a viable cell with non-functional mitochondria. Together, these results suggest the hypothesis that early expression of RSM22 may provide the methylation activity necessary for assembly of the mitochondrial ribosome.

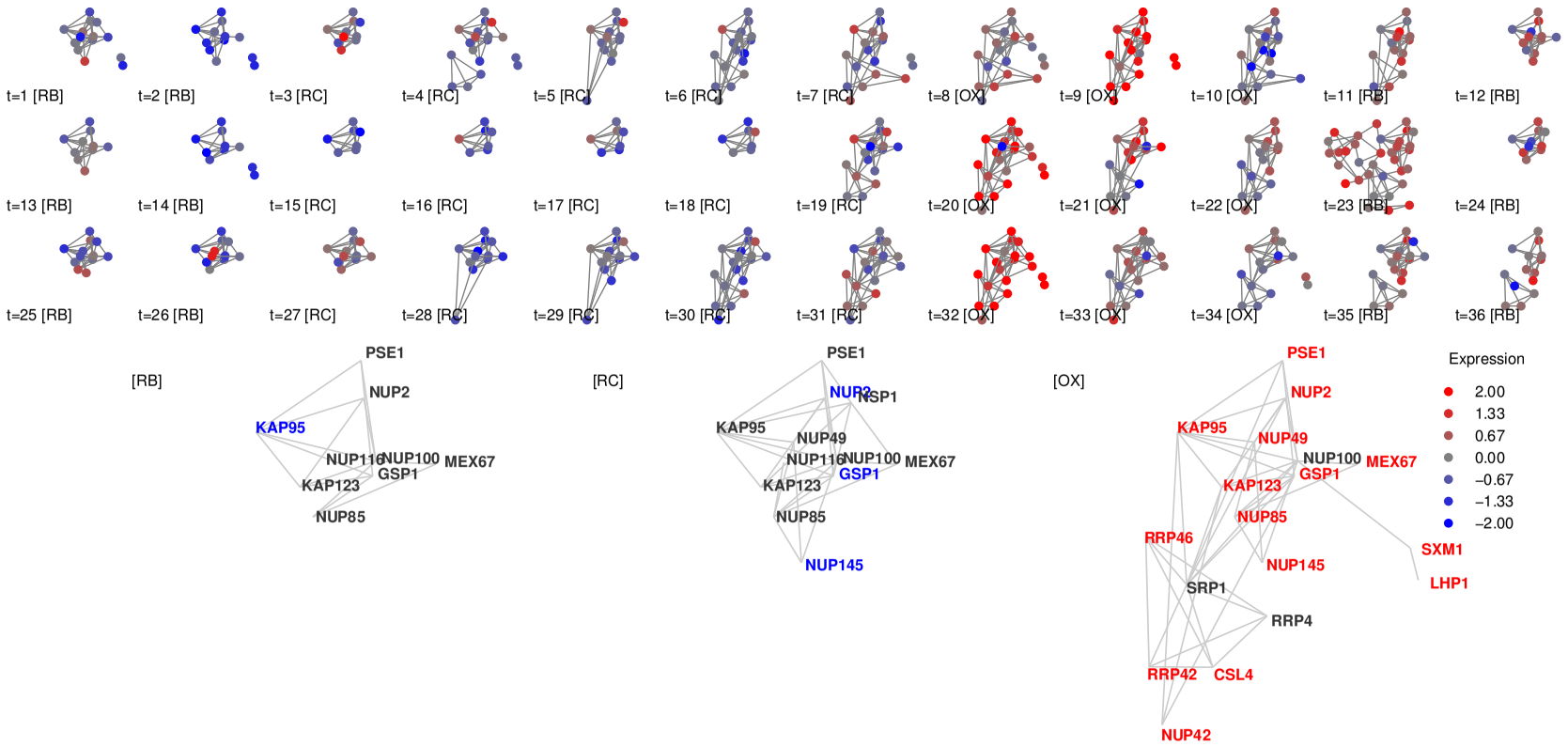

Figure 4.5: Cluster #16, nuclear pore complex. *Top:* cluster members for the 32 gene expression snapshots. *Bottom:* Average expression for the 3 YMC phases. *Node color:* standardized gene expression level. Gene names were colored red or blue if expression values are above 0.5 or -0.5 respectively.

Nuclear pore complex (#16) Most genes in the nuclear pore complex are OX-responsive and the complex is most active at $t = 9, 20, 32$ (Fig. 4.5). Unlike the mitochondrial ribosome, where the entire complex is generally transcribed in synchrony, this complex shows a smaller co-expressed core that is complemented with transient members during the OX phase. While it combines subunits of several annotated complexes, it has poor overlap with any single complex. Its best overlap is a 15% Jaccard correlation with high-throughput complex CID15 from YHTP2008.

The co-expressed core includes nuclear pore complex (NPC) and Karyopherin (KAP) proteins [146, 175]. The physical structure of the NPC comprises mostly NUP proteins. Among the proteins included in cluster #16, NUP2, NUP100, and NUP116 shape the Phe-Gly passage of the NPC [175]. In contrast, KAP proteins are not considered structural but rather mediate export and import of RNA and proteins [62, 175]. KAP123 and PSE1 specifically transport ribosomal proteins [165]. During the OX phases, SRP1 and SXM1 are additionally recruited. These KAP proteins recognize either nuclear localization sequences (NLS) or nuclear export sequences (NES) and direct transport into or out of nucleus [146].

Other transient memberships suggest additional hypotheses. RRP4 and RRP42 are a part of the exosome that edits RNA molecules $3' \rightarrow 5'$ [124]. Our clustering predicts that these proteins transition between the nuclear pore and other complexes during the cycle. CSL4 was recently reported to interact with RNA and is a possible exosome component [110]. LHP1 is a La protein that binds to RNA polymerase III transcripts and small ribonuclear proteins (snRNPs), working as a molecular chaperone to protect and terminate the $3'$ end of transcripts [200]. These results are consistent with the hypothesis that RNA processing is

tightly coupled to transport through the nuclear pore to the cytoplasm [175], but also suggest that dynamic reorganization of the nuclear pore occurs during the metabolic cycle. Additional evidence is the appearance of a second expression peak involving a subset of nuclear pore components at the start of the RB phase, which has not been previously described.

4.6 Arabidopsis root development

Dynamic biological network. The root is an ideal model for development because temporally staged samples are easily obtained by cutting further back from the root tip, and distinct cell and tissue types are observed radially outward from the root center (Fig. 4.6A). A classic study mapped gene expression activity in 5 spatial regions across 3 developmental stages [20], yielding 15 spatiotemporal snapshots.

High-confidence interactions for the corresponding proteins (confidence value ≥ 10) were extracted from TAIR Interactome 2.0 [52]. For this superposition of all genes active anywhere in the root map, we iteratively deleted network vertices with degree less than or equal to 3 until no more vertices could be removed. The resulting network had 332 vertices and 1163 edges. Subnetworks were then generated by extracting the active genes (expression level ≥ 75 as reported by [20]) and their interactions for each of the 15 snapshots. Each snapshot had approximately 150 to 220 genes and 5 interactions per gene (Table. 4.2).

Model selection. The depth of the hierarchical tree was set to 6 (64 groups). Results for occupied groups were substantially unchanged for depth-7 trees (128 groups, results not shown). DYHM introduces 8 spatiotemporal couplings with

	Stele	Endoderm	Endo+Cortex	Epiderm	Lateral root cap
Stage 3	217 (569)	215 (565)	225 (603)	219 (586)	211 (543)
Stage 2	182 (415)	185 (432)	193 (462)	188 (440)	172 (391)
Stage 1	150 (328)	151 (331)	156 (354)	144 (324)	135 (285)

Table 4.2: The spatiotemporal variation of active subnetworks. The numbers of active genes at each position are shown without parentheses; the numbers of active interactions are shown within the parentheses.

strength λ for adjacent tissues and stages (Fig. 4.6A). For the observed data $\mathcal{D}$ and a specific value of λ, we used a penalized likelihood to determine the degree of time-smoothness:

$$\mathcal{L}'(\mathcal{D}|\lambda) \;\; = \;\; \mathcal{L}(\mathcal{D}|\lambda) \times K!(K_T - K)!/(K_T + 1)!.$$

With M total groups (here 64), a total of $M(M-1) \equiv K_T$ directed transitions are possible. Of these, a subset K are observed at least once across the 8 coupled snapshots. The penalty $K!(K_T - K)!/(K_T + 1)!$ gives equal weight to each of the $C(K_T, K)$ models with exactly K transitions, which results in a steeper penalty for models with more transitions. This penalty arises from a Bayesian viewpoint in which each of the K_T possible transitions is observed independently with probability θ. Integrating $\int_0^1 \theta^K (1-\theta)^{K_T - K} d\theta$ produces the stated form of the penalized likelihood. We performed a search over a sparse grid, $\lambda = 0.01, 0.05, 0.1, 0.2$, and selected $\lambda = 0.1$ as the optimal value.

Hierarchical clustering and spatiotemporal mapping. Dynamical clustering using DYHM produces hierarchical cluster assignments for each of the 15 spatiotemporal samples. A reduced view of the results, averaging the inferred memberships over the 15 samples, is provided (Fig. 4.6B). The node color represents the averaged interaction enrichment. Leaf nodes, shaped as squares, are groups

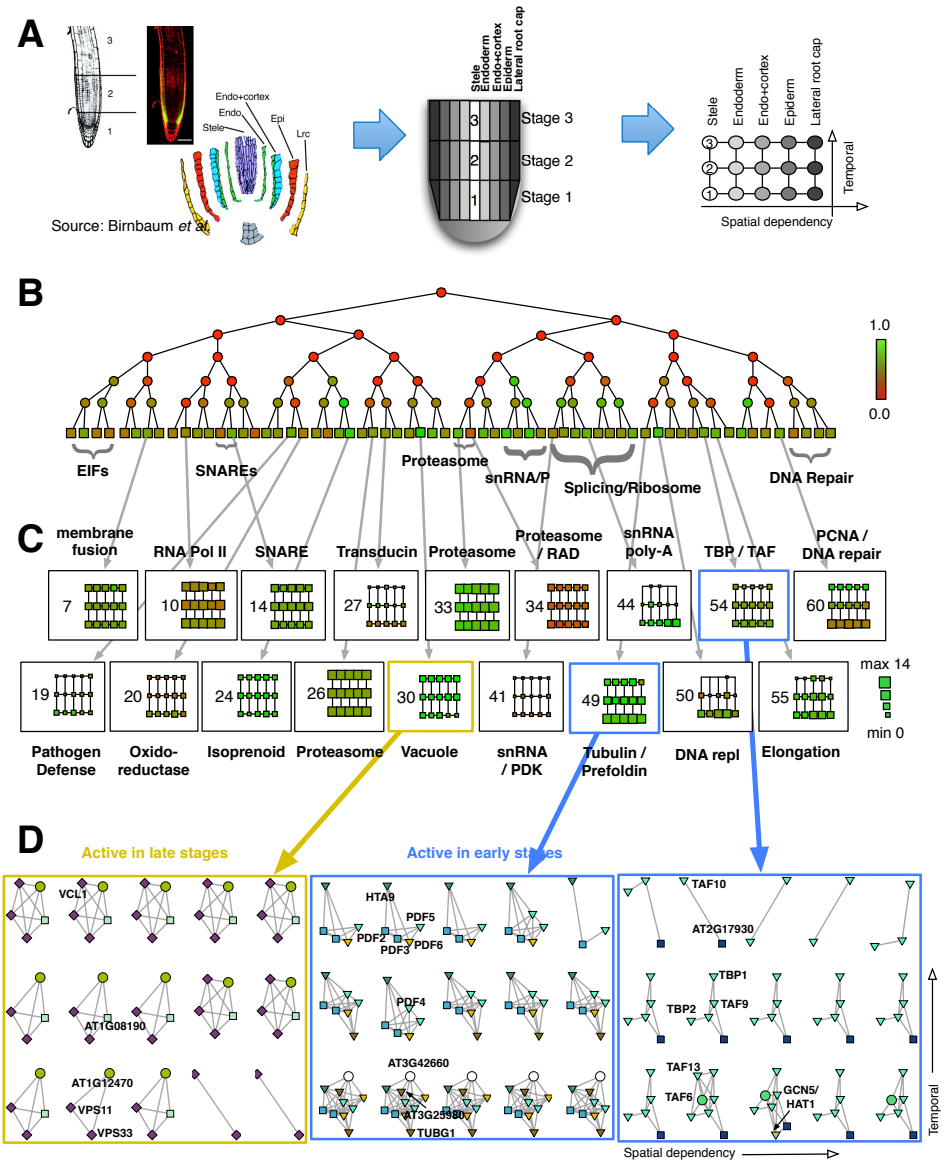

Figure 4.6: Arabidopsis root development. (A) Lateral root sections correspond to distinct tissues, and vertical sections correspond to to distinct developmental stages. (B) Average hierarchical decomposition of 15 networks. Node color indicates enrichment (green) or depletion (red) of within-cluster (at terminal nodes) or between-cluster (at internal nodes) edges relative to random connectivity. (C) The evolution of each cluster is displayed over the 5 tissues and 3 stages. Size indicates the number of proteins within the cluster, and color indicates edge enrichment. (D) Selected micro-views on network dynamics.

of clustered genes. These leaves are indexed from 1 (leftmost) to 64 (rightmost) for later reference. Zoomed-in views below illustrate how selected clusters evolve over space and time in increasing resolution (Fig. 4.6C,D).

This tree view shows that most of the groups are assortative (green nodes, enriched for self-interactions), which is typical of protein complexes. Some leaf nodes assemble hierarchically into larger assortative modules, and these components often share similar biological functions. For instance, four of small nuclear RNA/RNP complexes (snRNA/P) are located adjacently and form a clade (terminal leaves #39-40). Cladistic assignments are also observed for EIF (eukaryotic translation initiation factor) complexes (leaves #1-4) and Splicing/Ribosome complexes (leaves #41-48).

An overview of terminal groups shows how each of the 64 clusters varies over the 15 spatiotemporal snapshots in terms of occupancy and within-cluster interactions (Fig. 4.6C). Several of the clusters correspond to protein complexes that appear constitutively active, whose transcripts would typically be filtered out as unchanging. Examples are #7 (membrane fusion), #10 (RNA Pol II), #14 (syntaxin and SNARE proteins), and #26 and #33 (proteasome). A more dynamic pattern is observed for clusters that are conditionally activated, most often with complex members present at early times and then absent at later times to yield a smaller core complex. Examples are #44 (mRNA polyadenylation), #49 (a core of prefoldin and the H2A.Z histone variant HTA9 has additional tubulin-related complex members during stage 1), and #60 (a PCNA DNA repair complex is present in stage 1 but vanishes in stages 2 and 3). These observations are consistent with the inference from mRNA data of rapid mitotic activity during stage 1 [20].

TATA box-binding protein complex. A detailed view of cluster #54, involved in transcription from TATA box promoters, highlights this pattern of dynamic complex membership (rightmost of Fig. 4.6D). TATA box-binding protein associated factors (TAFs) have time-specific and tissue-specific activity [179]. One member of the TAF family, TAF10 (i.e., AT4G31720, TFIID15), has preferential and transient expression during the middle developmental stages of plant organs. Disrupting this tight regulation causes pleiotropic phenotypic changes and abnormal morphologies [179].

The majority of the genes in cluster #54 are TAFs, including TAFII15/TAF10, TAFII21/TAF9, and TAFII59/TAF6. In the root expression map, TAF10 is a core member of this complex, while other members are transient. Along the temporal axis, the TAF10-TAF9-TFIID-1 complex is present during early root development, persists partially through stage 2, and in the mature root only TAFII15, TBP2, and the uncharacterized PIK-related kinase AT2G17930 remain. TAFs provide DNA-binding specificity for TFIIDs, which bind to the basal transcriptional machinery [101]. The TAF6 (TAFII59) protein appears to be present primarily in stage 1, although absent from the stele. This factor has a core interaction motif required for H3/H4 heterodimerization [101], which suggests regional epigenetic modification in early development. At the early stage, this complex also has HAT1 as a member, a histoneacetyltransferase that is a positive regulator of transcription in root morphogenesis.

4.7 Biological impact

We have presented new methods for modeling the spatiotemporal dynamics of a biological network. The model takes as input a series of discrete network states coupled in space and time and infers a structure of dynamic groups that enter and leave the network, possibly merging or separating from existing groups.

Applied to a biological data set obtained from *Arabidopsis* root development, the model reveals the dynamic organization of network components. Previous analysis of this mRNA data set was limited to time-varying and spatially-varying genes. Of the roughly 22,000 transcripts interrogated, a half of them were not expressed in the root, a quarter of them showed differential regulation over space and time, and the remaining quarter of them were expressed constitutively. These unchanging transcripts are filtered out by traditional gene expression analysis.

For our analysis, the activity of each network component is inferred from transcript profiling, and the set of possible interactions is obtained from a database compendium. Our dynamic network model reveals that the constitutive components form the core of complexes that evolve through the addition and subtraction of dynamic modules. We are also able to observe modules that are strictly limited to specific spatiotemporal states and vanish elsewhere.

4.8 Technical impact

Converting real-valued gene expression levels to a binary presence/absence score for a protein is admittedly problematic. First, protein levels do not necessarily track mRNA levels. Second, the level of protein activity may not be adequately represented by a binary 0/1 score. We adopted this approach in part because it

was used in the original study. Given the promising performance of our initial application, further work may benefit by incorporating quantitative measures of gene or protein activity.

When we designed the DHAC algorithm, a Markov type of time-dependency was of our initial consideration, similar to a Markov chain of static exponential random graph model [65]. We found that kernelization, used previously in the KELLER algorithm for transcriptional networks [172], provides better performance. In contrast, the Markov chain approach performed worse than DHAC and only slightly better than HAC (results not shown) and is not included in the comparison. The Markov chain approach was not applicable in agglomerative clustering framework since initially contribution from temporal dependency is most dominating, and can mislead merging steps. More importantly both maximum likelihood and Bayesian estimation not necessarily favor strong diagonal terms in the transition matrix, therefore inference algorithm could choose either strong smoothness or strong discontinuity.

Time-constrained variational inference could extend in many other ways. Here we mainly constrained distance between two adjacent variational distributions of latent variables, but could generalize so that the distributions of higher-level parameters are coupled across time points [22]. However our main concern was more on the speed of the algorithm so we omitted additional spatiotemporal dependency structures. Nonetheless it is interesting direction and would be beneficial to future research.

The models we have introduced can be readily generalized to incorporate other time-dependent edge types, such as protein-DNA regulatory interactions or protein-protein modifications. Time dependence in the model described is

limited to time-varying module membership, but patterns of module-module interaction are held constant. As an analogy, consider a model of a citation network where patterns of citation by an author depend on the author's research group. In this model, a graduate student will follow the pattern of his or her PhD mentor, and then will take on the pattern of his or her postdoctoral mentor. The patterns of the mentors' groups remain fixed, however. In a more general model, the pattern for each mentor can itself evolve. This more general model is also amenable to an efficient variational optimization.

Chapter 5

Prioritization of network modules with contextual omics data sets

5.1 Introduction

Static and dynamic network clustering methods identify subnetworks, or modules or sets of vertices, that constitute an overall network as building blocks. A state-of-the-art study suggests that network modules are limited a constant size [108]. We reaffirmed that average sizes of modules in various sizes of biological networks ranges 5 to 10, therefore the number of modules increases with the number of vertices. For instance we normally identify thousands of modules in a large physical interaction map.

Gene set enrichment analysis (GSEA) and related algorithms identify gene sets that are enriched for genes, primarily based on p-value or differential expression fold-ratio or z-score. Examples include the original K-S statistic [176], difference in means [40, 113], difference in $|z|$ score [79]. Tests for significance of enrichment correspond to tests for a difference in parameters or distributions estimated for an underlying generative model. An alternative approach is to identify gene sets that are effective for discriminative learning of biological states. Here we develop new discriminative learning algorithms for paired / unpaired time-

course data and demonstrate that these methods perform better than competing enrichment-based approaches. Here, terms "gene set" and "module" refer to a set of genes, and we will use both interchangeably.

Discriminative learning. Suppose we have time-series $n \times T$ gene expression data for n genes at T time points for two biological states, with expression data denoted X (control, $\theta = 1$) and Y (case, $\theta = 0$). Each x_{it} and y_{it} measures gene i at time t. For simplicity we assume an average of biological or technical replicates if multiple measurement are available, and we have subtracted the mean so that $x_{it} = -y_{it}$. Genes in the set are modeled as independent, identically-distributed, and hence exchangeable observations. We want to measure discriminative power of each time point, not each gene. For a particular time point, a hyper-plane can be inferred by logistic regression with parameter β_t,

$$P(\theta_x = 1, \theta_y = 0 | x_{it}, y_{it}) = P(\theta_x = 1 | x_{it}) P(\theta_y = 0 | y_{it}) = \sigma(\mu + \beta_t x_{it})\, \sigma(-\mu - \beta_t y_{it})\,,$$

where the sigmoid function $\sigma(\xi) = e^{\xi}/(1 + e^{\xi})$ and $\mu = 0$ for mean-subtracted data. The parameter $\{\beta_t\}$ is estimated by maximizing the likelihood function

$$\mathcal{L}(\theta_x = 1, \theta_y = 0, X, Y; \boldsymbol{\beta}) = \prod_{i,t} \sigma(\beta_t x_{it})\, \sigma(-\beta_t y_{it})\,. \tag{5.1}$$

For each time point, the genes in the set share the parameter β_t, which quantifies the discriminative power of the gene set at that time point. We term this model Temporal Expression Divergence, or TED.

Paired time-course data. Many methods "generalize" to time-course data, but often this means that time points are treated as indistinguishable examples without no real ordering. Discrimination based on a sum over the trajectory, such

as $\sigma(\sum_t \beta_t x_{it})\,\sigma(-\sum_t \beta_t y_{it})$, is possible but sensitive to systematic biases such as batch effects [105], which can then dominate the time course; it also ignores ordering. TED treats each time point separately, similar to product of experts [120], rather than mixture of experts. We account for time-ordering by enforcing temporal smoothness, implemented using fused lasso regularization [181].

Multiple unpaired data. However, we often need face situations where direct pairing of two or more arrays is formidable. Then possibly necessary assumptions would include that times are statistically identical. It appears so in our angiogenesis application, since different sets of time-course data are measured at different scale. We build upon the TED and design a new model that discriminates multiple conditions in all-pairwise comparisons.

Suppose we have T types of gene expression arrays, but different types have different number of samples. We denote a set S_t to refer to sample indexes that belong to type t, and m_t be size of the set S_t. Let X be total expression arrays of n genes and $(\sum_{t=1}^{T} m_t)$ samples. A vector $\mathbf{x}_i$ of gene $i \in [n]$ then partitions into T expression sets $\mathbf{x}_i^{(t)} \equiv \{x_{ij} : j \in S_t\}$, $t \in [T]$.

In all-pairwise comparison of types, $\{a < b\}$, we can arbitrary assign labels either 0 or 1. For each gene i we fit a two-parameter logistic regression model, which takes the likelihood function

$$\prod_{a<b} \mathcal{L}_{ab}(\theta_a = 1, \theta_b = 0, \mathbf{x}_i^{(a)}, \mathbf{x}_i^{(b)}; \boldsymbol{\beta}^{(a,b)}), \tag{5.2}$$

where

$$\mathcal{L}_{ab} = \prod_{j \in S_a} \sigma\left(\beta_0^{(a,b)} + \beta_1^{(a,b)} x_{ij}\right) \prod_{j' \in S_b} \sigma\left(-\beta_0^{(a,b)} - \beta_1^{(a,b)} x_{ij'}\right).$$

We term our approach, multi-way expression divergence model, or MED. The

magnitude of $\beta_1^{(a,b)}$ parameters determines the steepness of logistic functions, or decision boundary, by which we quantify discriminative power between pairs.

Multi-modality Gene sets are lists of genes in same pathways and complexes. In practice, gene sets are often large and contain both up-regulated and down-regulated genes, which may correspond to activators or repressors. Many methods perform separate tests of up-regulated and down-regulated genes by truncation [176] or "max mean" statistics [40], or perform an overall test using χ^2 statistics that measure effect magnitude ignoring direction [79]. We pursue an alternative principled approach, Dirichlet Process Mixture (DPM) models, that allow the discriminative model to fit an unknown number of modes using a non-parametric Bayesian prior [5]. For each observation i we define random variable $c_i \in [K]$, with $K \to \infty$, to indicate membership in K TED models. Each TED model k is parameterized by $\boldsymbol{\beta}^{(k)}$. We term this approach TED-dpm.

$$\mathcal{L}(X, Y; \{\boldsymbol{\beta}^{(k)}\}) = \prod_{k=1}^{K} \mathcal{L}(\mathbf{x}_i, \mathbf{y}_i; \boldsymbol{\beta}^{(k)})^{\mathbb{1}[c_i=k]}. \tag{5.3}$$

The DPM prior favors distributions that concentrate genes into a small number of modes, with the potential to refine gene sets into discriminative sub-sets and discarding uninformative genes. For the MED model, we apply the same strategy and term the method MED-dpm.

5.2 Temporal expression divergence

Discriminative learning algorithm

Bayesian inference and hypothesis testing We do Bayesian inference on unknown parameters $\boldsymbol{\beta}$ of TED model (Eq. 5.1). By asymptotic normality, $(\beta_t - \mathbb{E}[\beta]_t)/\sqrt{\mathbb{V}[\beta]_t} \approx \mathcal{N}(0,1)$. We perform a Wald test of $H_0 : \beta_t = 0$ versus $H_1 : \beta_t \neq 0$. We reject H_0 at level α if observed

$$|z^\star| = \left| \frac{\mathbb{E}[\beta_t]}{\sqrt{\mathbb{V}[\beta_t]}} \right| > \Phi^{-1}(1-\alpha/2),$$

where $\Phi(\cdot)$ is the cumulative normal distribution. Similarly we compute

$$\text{p-value} = \Pr\left(|Z| > |z^\star|\right), \quad \text{with } Z \sim \mathcal{N}(0,1). \tag{5.4}$$

In a strict sense our hypothesis testing is akin to controlling false discovery rate [39, 177, 178], rather than Type I error. Therefore we used Holm's procedure [77, 106] for multiple hypothesis correction, even for controlling false discovery rate.

Bayesian sparse prior We use a fused Lasso prior [181] for the $\boldsymbol{\beta}$ parameters,

$$P(\boldsymbol{\beta}) \propto \exp\left(-\frac{\lambda}{2} \sum_{t=1}^{T} |\beta_t| - \frac{\gamma}{2} \sum_{t=1}^{T-1} |\beta_{t+1} - \beta_t| \right), \tag{5.5}$$

where λ and γ control static and kinetic sparsity of parameters. Instead of direct optimization [48], we used the equivalent Bayesian Lasso [100, 139],

$$\begin{aligned}
\beta_j|\tau &\sim \mathcal{N}\left(\beta_j \middle| 0, \tau_j^2\right), \\
\tau_j^2 &\sim \mathsf{Exp}\left(\tau_j^2 \middle| \lambda/2\right), \\
\beta_j - \beta_{j+1}|\kappa &\sim \mathcal{N}\left(\beta_j - \beta_{j+1} \middle| 0, \kappa_j^2\right), \\
\kappa_j^2 &\sim \mathsf{Exp}\left(\kappa_j^2 \middle| \gamma/2\right).
\end{aligned}$$

Non-conjugate variational inference To side-step the non-conjugate relationship between the likelihood function (Eq. 5.1) and prior (Eq. 5.5), we perform posterior inference by the non-conjugate variational method [191]. We find the optimal $\boldsymbol{\beta}$ first, then construct an approximate distribution $Q(\boldsymbol{\beta})$ by the Laplace method. We derived coordinate descent steps [49] that minimize

$$f(\boldsymbol{\beta}) = -\mathbb{E}[\log \mathcal{L}(X, Y; \boldsymbol{\beta})] - \mathbb{E}[\log P(\boldsymbol{\beta}|\tau, \kappa)]$$

iteratively. We construct a local quadratic form $g(\boldsymbol{\beta}) \approx f(\boldsymbol{\beta})$ at current estimate $\hat{\boldsymbol{\beta}}$, defined by

$$\begin{aligned} g(\boldsymbol{\beta}) &= \frac{1}{2}\sum_{t=1}^{T}\sum_{i=1}^{n} w_{it}(\beta_t - v_{it})^2 \\ &\quad + \frac{1}{2}\sum_{t=1}^{T}\mathbb{E}\left[\frac{1}{\tau_t^2}\right]\beta_t^2 + \frac{1}{2}\sum_{t=1}^{T-1}\mathbb{E}\left[\frac{1}{\kappa_t^2}\right](\beta_t - \beta_{t+1})^2 \end{aligned}$$

where

$$\begin{aligned} v_{it} &= \hat{\beta}_t + \frac{x_{it}\sigma(-x_{it}\hat{\beta}_t) - y_{it}\sigma(y_{it}\hat{\beta}_t)}{w_{it}}, \\ w_{it} &= x_{it}^2\sigma(x_{it}\hat{\beta}_t)\,\sigma(-x_{it}\hat{\beta}_t) + y_{it}^2\sigma(y_{it}\hat{\beta}_t)\,\sigma(-y_{it}\hat{\beta}_t)\,. \end{aligned}$$

We can optimize $g(\boldsymbol{\beta})$ with respect to $\boldsymbol{\beta}$ in a closed form solution by choosing a stationary point. For each time point $t \in [T]$ we iteratively update $\hat{\beta}_t$ until convergence,

$$\hat{\beta}_t \leftarrow \frac{\sum_{i=1}^{n} w_{it}v_{it} + \hat{\beta}_{t-1}\mathbb{E}\left[1/\kappa_{t-1}^2\right] + \hat{\beta}_{t+1}\mathbb{E}\left[1/\kappa_t^2\right]}{\sum_{i=1}^{n} w_{it} + \mathbb{E}\left[1/\tau_t^2\right] + \mathbb{E}\left[1/\kappa_{t-1}^2\right] + \mathbb{E}\left[1/\kappa_t^2\right]}, \tag{5.6}$$

dropping the $\mathbb{E}\left[1/\kappa_{t-1}^2\right]$ term for $t = 1$ and $\mathbb{E}\left[1/\kappa_{t+1}^2\right]$ for $t = T$.

Let $\hat{\boldsymbol{\beta}}$ be optimal solution. Setting $\mathbb{E}[\boldsymbol{\beta}] = \hat{\boldsymbol{\beta}}$ and the precision $\Lambda = \nabla^2 f(\hat{\boldsymbol{\beta}})$ (see Wang and Blei for justification [191]), we approximate

$$Q(\boldsymbol{\beta}|\cdot) \approx \mathcal{N}\left(\boldsymbol{\beta}\middle|\hat{\boldsymbol{\beta}}, \Lambda^{-1}\right). \tag{5.7}$$

Elements of the tridiagonal precision matrix Λ are

$$\begin{aligned}\Lambda_{tt} &= \sum_{i=1}^{n} w_{it} + 1/\tau_t^2 + 1/\kappa_{t-1}^2 + 1/\kappa_t^2, \quad t \in [T], \\ \Lambda_{t,t+1} &= \Lambda_{t+1,t} = -1/\kappa_t^2, \quad t \in [T-1].\end{aligned}$$

Distribution of τ and κ The posterior distributions of $1/\tau_t^2$ and $1/\kappa_t^2$ are inverse normal based on a general result [100]. We can resolve the mean-field solution required for the update of $Q(\boldsymbol{\beta})$ and the empirical Bayes estimation penalty terms as

$$\mathbb{E}\left[1/\tau_t^2\right] = \sqrt{\frac{\lambda^2}{\mathbb{E}\left[\beta_t^2\right]}}, \quad \mathbb{E}\left[1/\kappa_t^2\right] = \sqrt{\frac{\gamma^2}{\mathbb{E}[(\beta_t - \beta_{t-1})^2]}} \tag{5.8}$$

and

$$\mathbb{E}\left[\tau_t^2\right] = \sqrt{\frac{\mathbb{E}\left[\beta_t^2\right]}{\lambda^2}} + \frac{1}{\lambda^2}, \quad \mathbb{E}\left[\kappa_t^2\right] = \sqrt{\frac{\mathbb{E}[(\beta_t - \beta_{t-1})^2]}{\gamma^2}} + \frac{1}{\gamma^2}, \tag{5.9}$$

where

$$\mathbb{E}\left[\beta_t^2\right] = \hat{\beta}_t^{\,2} + \Sigma_{tt}, \quad \mathbb{E}\left[(\beta_t - \beta_{t+1})^2\right] = \mathbb{E}\left[\beta_t^2\right] + \mathbb{E}\left[\beta_{t+1}^2\right] - 2(\hat{\beta}_t\hat{\beta}_{t+1} + \Sigma_{t,t+1}). \tag{5.10}$$

The tri-diagonal elements of $\Sigma \equiv \Lambda^{-1}$, i.e., Σ_{tt} and $\Sigma_{t,t+1}$ are calculated using linear time algorithms [163, 197].

Empirical Bayes We adjust the penalty parameters, λ and γ, by optimizing the marginal likelihood weighted by the variational distributions over $\boldsymbol{\beta}$ and τ, κ [100, 139]. We update λ and γ as

$$\frac{1}{\lambda^2} \leftarrow \frac{\sum_{t=1}^{T} \mathbb{E}\left[\tau_t^2\right]}{2T}, \quad \frac{1}{\gamma^2} \leftarrow \frac{\sum_{t=1}^{T-1} \mathbb{E}\left[\kappa_t^2\right]}{2(T-1)}. \tag{5.11}$$

until convergence.

Overall inference algorithm. We summarize overall inference algorithm in Alg. 7.

Alg 7 TED

$\mathbb{E}\left[1/\tau_t^2\right] \leftarrow 0$ for all $t \in [T]$
$\mathbb{E}\left[1/\kappa_t^2\right] \leftarrow 0$ for all $t \in [T-1]$
repeat
  repeat
    optimize $\hat{\boldsymbol{\beta}}$ (Eq. 5.6)
    update $Q(\boldsymbol{\beta}|\cdot) \approx \mathcal{N}\left(\boldsymbol{\beta}\middle|\hat{\boldsymbol{\beta}}, \Lambda^{-1}\right)$ (Eq. 5.7)
    calculate covariance matrix $\Sigma \equiv \Lambda^{-1}$
    estimate $\mathbb{E}\left[\beta_t^2\right]$ and $\mathbb{E}\left[(\beta_t - \beta_{t+1})^2\right]$ (Eq. 5.10)
    update $\mathbb{E}\left[1/\kappa_t^2\right], \mathbb{E}\left[1/\tau_t^2\right]$ (Eq. 5.8)
  until convergence of τ, κ
  repeat
    update of $\mathbb{E}\left[\tau^2\right], \mathbb{E}\left[\kappa^2\right]$ (Eq. 5.9)
    empirical Bayes estimation of λ, γ (Eq. 5.11)
  until convergence of λ, γ
until convergence of overall result

Dirichlet process mixture of TED

Locally collapsed latent update We assign a pair of expression vectors $\mathbf{x}_i$ and $\mathbf{y}_i$ to K TED models in a mixture or admixture framework. Membership variable $z_i \in [K]$ represents model assignment, and $K \to \infty$ for DPM. Each TED model k is characterized by $\hat{\boldsymbol{\beta}}_k$ and Λ_k (Eq. 5.7). We derive latent assignment update via "locally collapsed" variational inference (LCVI) [190]. We locally sample $z_i = k$ with probability

$$Q(z_i = k|\cdot) \propto \Pr(c_i = k) \prod_{t=1}^{T} \exp\left(f(\hat{\beta}_{kt}) + \frac{1}{2}\frac{f'(\hat{\beta}_{kt})^2}{\Lambda_{ktt} - f''(\hat{\beta}_{kt})}\right) \sqrt{\frac{\Lambda_{ktt}}{\Lambda_{ktt} - f''(\hat{\beta}_{kt})}} \tag{5.12}$$

where β_{kt} denotes t-th element of $\boldsymbol{\beta}_k$ vector; Λ_{ktt} denotes (t,t) element of Λ_k matrix. We also calculate the appropriate prior factor $\Pr(z_i = k)$.

For a model with no previous observations, we simply treat $\beta_t \sim \mathcal{N}(\beta_t|0,\tau^2)$ with $1/\tau^2 = \infty$; we use the result $1/\tau^2 \approx \lambda/|\beta_t| \to \infty$ as $\beta \to 0$.

Dirichlet Process Mixture First we randomly assign expression pairs $(\mathbf{x}_i, \mathbf{y}_i)$ to models and update the models given data. For each observation, we sample assignments of pairs (Eq. 5.12) analogous to the E-step of Expectation Maximization [34], then perform variational update of each TED model including empirical Bayes routines analogous to an M-step [190]. Overall procedure is summarized in Alg. 8.

Alg 8 TED-dpm

```
for i ∈ [n] do
    uniform sampling of z_i from [K]
end for
repeat
    probabilistically assign data points to each k component
    inference of each TED model (Alg. 7)
    for i ∈ [n] do
        estimate Q(z_i) by collapsed Gibbs sampling (Eq. 5.12)
    end for
until convergence of overall algorithm
```

Performance evaluation

We compared performance of TED and TED-dpm with existing methods. We include gene set analysis GSA [40] because it had generally good performance in previous experiments [115]. We also compare with Generally Applicable Gene

set Enrichment adaptive paired/unpaired sample comparison (GAGE) [112,113], which also have performed well. While a more comprehensive comparison could be beneficial, these methods are fast, robust, applicable to paired time point data, and have performed well in head-to-head comparisons with real data.

We simulated case-control study with 10 time points, 100 gene sets, and genes per gene set chosen uniformly from $\{20, 50, 70, 100, 150\}$. Genes in gene sets at time points with no difference between case and control follow a null distribution, $x_{it} \sim \mathcal{N}(0,1)$ and $y_{it} \sim \mathcal{N}(0,1)$, where $\mathcal{N}(\mu, \sigma^2)$ is a normal distribution with mean μ and variance σ^2. For each simulation, 10 of the 100 gene sets were selected to have expression differences at either the first 4 or the first 7 time points, with half of the genes in the set having differential expression and the other half following the null. For genes with an expression difference, $x_{it} \sim \mathcal{N}(0.5, 1)$ and $y_{it} \sim \mathcal{N}(-0.5, 1)$. This process was repeated 10 times. GAGE and GSA output p-values and adjusted p-values directly. In TED and TED-dpm we generate a z-scores for each gene set at each time point, $z_t = \beta_t / \sqrt{\mathbb{V}[\beta_t]}$, which are then aggregated to calculate a p-value as $2(1 - \Phi(|\tilde{z}|))$ where $\tilde{z} = \sum_{t=1}^{10} z_t / \sqrt{10}$. The empirical false discovery rate (FDR) and power at significance cutoff $c \in [0,1]$ are

$$\text{FDR} = \frac{|\{i \in [m] : p_i \leq c \wedge \phi_i = 0\}|}{|\{i \in [m] : p_i \leq c\}|}, \quad \text{power} = \frac{|\{i \in [m] : p_i \leq c \wedge \phi_i = 1\}|}{|\{i \in [m] : \phi_i = 1\}|}. \tag{5.13}$$

We evaluated prediction accuracy by plotting FDR on the x-axis and power on the y-axis, equivalent to a precision-recall curve. Cutoff values c were determined by sorted order of p-values.

The TED-dpm method dominates the other methods (Fig. 5.1). When only

40% of time points are informative, TED and TED-dpm have higher power at the empirical FDR (Fig. 5.1a). Even when information is high, with 70% of time points informative, power at controlled FDR is consistently higher for TED-dpm than for other methods (Fig. 5.1b).

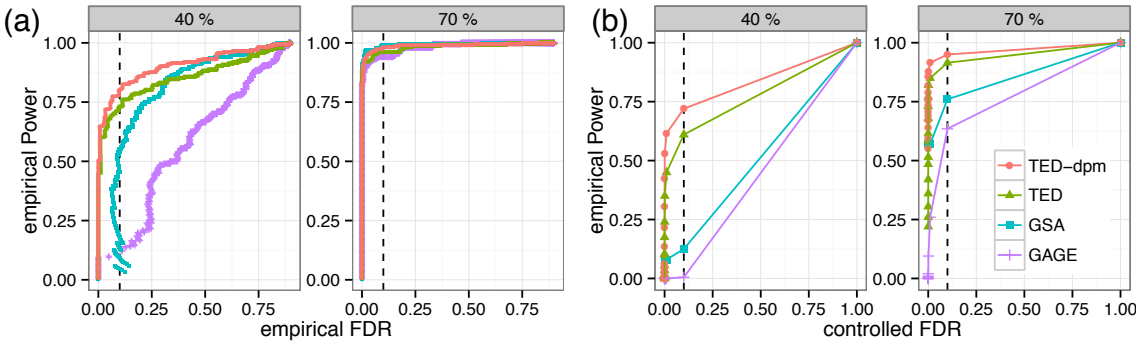

Figure 5.1: Empirical power comparison on simulated data sets. Shaded titles indicate the fraction of informative time points, 40% or 70%. (a) Comparison of prediction accuracy based on ranked order by p-values; (b) Comparison of statistical power at controlled FDR.

5.3 Multi-way expression divergence

Discriminative learning algorithm

Main idea is the same as the TED model. We estimate posterior distribution of parameters $\boldsymbol{\beta}^{(a,b)}$ all-pairwise, and test significance of the parameters, and determine diverging gene sets in multiple hypothesis testing.

Given the posterior distribution of $\boldsymbol{\beta}^{(a,b)}$ parameters, we calculate mean $\mathbb{E}\left[\beta_1^{(a,b)}\right]$ and variance $\mathbb{V}\left[\beta_1^{(a,b)}\right]$ of the slope parameters, and derive test statistics based on

theses sufficient statistics. The test statistic of a pair (a, b) is

$$Z_{ab} = \left| \frac{\mathbb{E}\left[\beta_1^{(a,b)}\right]}{\sqrt{\mathbb{V}\left[\beta_1^{(a,b)}\right]}} \right|,$$

and under the null hypothesis it asymptotically follows the standard Normal, i.e., $Z_{ab} \sim \mathcal{N}(0, 1)$. Given the observed test statistic $z^\star$, we calculate p-value

$$p^{(a,b)} = \Pr\left(|Z| > |z^\star|\right), \quad \text{with } Z \sim \mathcal{N}(0, 1). \tag{5.14}$$

We may combine all the pair of p-values, $\{p^{(a,b)}\}$, to have a p-value $p^{(a)}$ that measures significance for a specific type a.

$$p^{(a)} = \Pr\left(|Z| > |z_a|\right) \tag{5.15}$$

where

$$z_a = \sum_{b \neq a} F^{-1}(p^{(a,b)}), \quad F(x) = 1 - \Phi(x), \quad Z \sim \mathcal{N}(0, 1).$$

Note Φ denotes cumulative density function of the standard Normal distribution.

We also impose a sparse prior over the parameters:

$$P(\{\boldsymbol{\beta}^{(a,b)}\}) \propto \exp\left(-\frac{\lambda}{2} \sum_{a<b} \sum_{j=0}^{1} |\beta_j^{(a,b)}|\right), \tag{5.16}$$

which can be cast into Bayesian Lasso [139] priors:

$$\begin{aligned} \beta_j^{(a,b)} | \tau_j^{(a,b)} &\sim \mathcal{N}\left(\beta \middle| 0, {\tau_j^{(a,b)}}^2\right), \\ \tau_j^{(a,b)} &\sim \mathsf{Exp}\left({\tau_j^{(a,b)}}^2 \middle| \lambda/2\right). \end{aligned}$$

For each pair (a, b) we estimate β and τ iteratively until convergence, and then set the penalty parameter λ to an Empirical Bayes estimate based upon all-pairwise posterior distributions $P(\beta^{(a,b)}|\cdot)$.

Non-conjugate variational inference. Again, exact posterior inference is intractable because of non-conjugate relations between the priors (Eq. 5.16) and likelihood (Eq. 5.2). Therefore we use non-conjugate variational inference [191], that we find an optimal solution and estimate distribution around the optimum by Laplace approximation.

Optimization by coordinate descent. Since we treat the pairs of conditions are independent with each other we perform optimization and inference separately. Consider an arbitrary pair (a, b). We denote expressions of condition a by X and that of b by Y, i.e., $X = \bigcup_{i\in[n]} \mathbf{x}_i^{(a)}$ and $Y = \bigcup_{i\in[n]} \mathbf{x}_i^{(b)}$. Let the samples of a be A and b be B. The negative log likelihood function is

$$-\log \mathcal{L}_{ab}(X, Y; \beta) = \sum_i \left[\sum_{a\in A} \log(1 + e^{-\beta_0 - \beta_1 x_{ia}}) + \sum_{b\in B} \log(1 + e^{\beta_0 + \beta_1 y_{ib}}) \right],$$

with the prior distribution [139],

$$P(\beta) = \mathcal{N}\left(\beta_0 \middle| 0, \tau_0^2\right) \mathcal{N}\left(\beta_1 \middle| 0, \tau_1^2\right).$$

For a *maximum a posteriori* estimation of $\boldsymbol{\beta}^{(a,b)}$, we minimize

$$f(\beta) = -\log \mathcal{L}_{ab}(X, Y; \beta) - \log P(\beta).$$

However, this is analytically intractable, especially for the coordinate ascent algorithm [49]. Therefore we derive quadratic approximation of the objective,

$$f(\beta) \approx \sum_i g_i(\beta) + \frac{1}{2\tau_0^2}\beta_0^2 + \frac{1}{2\tau_1^2}\beta_1^2 + \text{const.} \tag{5.17}$$

where

$$\begin{aligned} g_i(\beta) &= \frac{1}{2}\sum_{a\in A} w_{ia}(\beta_0+\beta_1 x_{ia}-r_{ia})^2+\frac{1}{2}\sum_{b\in B} v_{ib}(\beta_0+\beta_1 y_{ib}+s_{ib})^2, \\ w_{ia} &= \sigma(\hat\beta_0+\hat\beta_1 x_{ia})\,\sigma(-\hat\beta_0-\hat\beta_1 x_{ia}), \\ r_{ia} &= \hat\beta_0+\hat\beta_1 x_{ia}+1+e^{-\hat\beta_0-\hat\beta_1 x_{ia}}, \\ v_{ib} &= \sigma(-\hat\beta_0-\hat\beta_1 y_{ib})\,\sigma(\hat\beta_0+\hat\beta_1 y_{ib}), \\ s_{ib} &= -\hat\beta_0-\hat\beta_1 y_{ib}+1+e^{\hat\beta_0+\hat\beta_1 y_{ib}}. \end{aligned}$$

Since it is quadratic we can easily find the solution by setting

$$\begin{aligned} \beta_0 &\leftarrow \frac{-\sum_i\sum_{b\in B} v_{ib}(s_{ib}+\beta_1 y_{ib})+\sum_i\sum_{a\in A} w_{ia}(r_{ia}-\beta_1 x_{ia})}{\sum_i\sum_{b\in B} v_{ib}+\sum_i\sum_{a\in A} w_{ia}+1/\tau_0^2}, \\ \beta_1 &\leftarrow \frac{-\sum_i\sum_{b\in B} v_{ib}(\beta_0+s_{ib})y_{ib}+\sum_i\sum_{a\in A} w_{ia}(r_{ia}-\beta_0)x_{ia}}{\sum_i\sum_{b\in B} v_{ib}y_{ib}^2+\sum_i\sum_{a\in A} w_{ia}x_{ia}^2+1/\tau_1^2}. \end{aligned} \tag{5.18}$$

We repeat the overall procedure, constructing quadratic approximation and solving the approximate objective, until convergence. Since the original objective $f(\beta)$ is also convex, we have guaranteed convergence.

Non-conjugate variational update Given the optimal $\boldsymbol{\beta}^{(a,b)}$, we estimate the precision matrix

$$\Lambda = \begin{bmatrix} -\partial^2 l/\partial\beta_0^2 & -\partial^2 l/\partial\beta_0\partial\beta_1 \\ -\partial^2 l/\partial\beta_0\partial\beta_1 & -\partial^2 l/\partial\beta_1^2 \end{bmatrix}, \tag{5.19}$$

where

$$\begin{aligned} \frac{\partial^2}{\partial\beta_0^2} l(\beta) &= -\sum_i\left[\sum_a w_{ia}+\sum_b v_{ib}\right], \\ \frac{\partial^2}{\partial\beta_1^2} l(\beta) &= -\sum_i\left[\sum_a w_{ia}x_{ia}^2+\sum_b v_{ib}y_{ib}^2\right], \\ \frac{\partial^2 l(\beta)}{\partial\beta_0\partial\beta_1} &= -\sum_i\left[\sum_a w_{ia}x_{ia}+\sum_b v_{ib}y_{ib}\right]. \end{aligned}$$

Terms w_{ia} and v_{ib} are calculated in the quadratic optimization (Eq.5.17). We also combine contribution from the prior probability on $\boldsymbol{\beta}$,

$$\Lambda_0 = \begin{bmatrix} 1/\tau_0^2 & 0 \\ 0 & 1/\tau_1^2 \end{bmatrix}. \tag{5.20}$$

Here we can simply set $\mathbb{E}[\boldsymbol{\beta}] = \hat{\boldsymbol{\beta}}$ and $\mathbb{V}[\boldsymbol{\beta}]^{-1} = \Lambda + \Lambda_0$, yielding approximate variational distribution $Q(\beta|\cdot) \approx \mathcal{N}(\beta|\mathbb{E}[\beta], \mathbb{V}[\beta])$, and this works well [191]. However we can derive more robust update inference algorithm by stochastic update [73]. Here, we gradually update

$$\Lambda^{(t+1)} \leftarrow \rho_t \Lambda + (1-\rho_t)\Lambda^{(t)}$$

and

$$\mu^{(t+1)} \leftarrow \rho_t [\hat{\beta}_0, \hat{\beta}_1]\Lambda + (1-\rho_t)\mu^{(t)}$$

with appropriate leaning rate $\rho_t \in (0, 1]$. We recover the posterior in terms of μ and Λ:

$$Q(\beta|\cdot) \approx \mathcal{N}(\beta|\mathbb{E}[\boldsymbol{\beta}], \mathbb{V}[\boldsymbol{\beta}]) \tag{5.21}$$

where

$$\mathbb{E}[\boldsymbol{\beta}] = \mu\,\Lambda^{-1}, \quad \mathbb{V}[\boldsymbol{\beta}] = (\Lambda + \Lambda_0)^{-1}.$$

Update of τ parameters We simply reuse update equations derived from static penalty of TED (Eq. 5.8 and Eq. 5.9), that is

$$\mathbb{E}\left[(1/\tau_t)^2\right] \leftarrow \sqrt{\frac{\lambda^2}{\mathbb{E}\left[{\beta_t}^2\right]}}, \quad \mathbb{E}\left[{\tau_t}^2\right] \leftarrow \sqrt{\frac{\mathbb{E}\left[{\beta_t}^2\right]}{\lambda^2}} + \frac{1}{\lambda^2} \tag{5.22}$$

for $t = 0, 1$. Unlike TED, we can easily compute 2×2 covariance matrix $\mathbb{V}[\boldsymbol{\beta}]$. We use its diagonal terms and find

$$\mathbb{E}\left[\beta_t^2\right] = \mathbb{E}[\beta_t]^2 + \mathbb{V}[\beta_t]$$

for $t = 0, 1$.

Empirical Bayes estimation of hyper-parameter We fix the lasso penalty λ by empirical Bayes estimate [139].

$$\frac{1}{\lambda^2} \leftarrow \frac{\sum_{a<b} \sum_{t=0}^{1} \mathbb{E}\left[{\tau_t^{(a,b)}}^2\right]}{4 \sum_{a<b} 1} \tag{5.23}$$

where we combine the estimated $\mathbb{E}\left[{\tau^{(a,b)}}^2\right]$ of all-pairwise relations.

Overall inference algorithm The learning algorithm for the MED is essentially the same as TED, , but we reiterate all the steps for completeness (Alg. 9).

Alg 9 MED

repeat
  repeat
    for $a < b$ **do**
      find optimal $\boldsymbol{\beta}^{(\hat{a},b)}$ by Eq. 5.18
      update the distribution $Q(\boldsymbol{\beta}^{(a,b)})$ by Eq. 5.21
      update the auxiliary $\mathbb{E}\left[(1/\tau^{(a,b)})^2\right]$ by Eq. 5.22
    end for
  until convergence of τ
  repeat
    estimate new penalty λ by Eq. 5.22
    reevaluate $\mathbb{E}\left[\tau^2\right]$ by Eq. 5.23
  until convergence of λ
until convergence of overall

Dirichlet process mixture of MED

Locally collapsed variational inference We present updates on an arbitrary pair (a, b), but other pairs are identical. For clear demonstration, let X be $n \times m_a$ matrix and Y be $n \times m_b$ matrix for conditions a and b. Rows correspond to n genes and columns correspond to samples/observations. For each gene i, we have m_a-vector $\mathbf{x}_i$ and m_b-vector $\mathbf{y}_i$.

The original probability is analytically intractable; therefore, we use the quadratic approximation (Eq. 5.17), that

$$P(\mathbf{x}_i, \mathbf{y}_i|\boldsymbol{\beta}) \approx \frac{1}{\mathcal{Z}} e^{\mathcal{G}_i(\boldsymbol{\beta})}$$

where $\mathcal{Z}$ normalizes the density function. We approximate the probability of latent assignment by integrating out the parameter [190]. Therefore we integrate out $\boldsymbol{\beta}$ with respect to the variational distribution (Eq. 5.21),

$$\begin{aligned} Q(\mathbf{x}_i, \mathbf{y}_i|\cdot) &\approx \frac{1}{\mathcal{Z}} \int \exp\{g_i(\boldsymbol{\beta})\} \mathcal{N}(\boldsymbol{\beta}|\mathbb{E}[\boldsymbol{\beta}], \mathbb{V}[\boldsymbol{\beta}])\, d\boldsymbol{\beta} \qquad (5.24) \\ &\propto \frac{1}{\sqrt{\det \mathbb{V}[\boldsymbol{\beta}]}\sqrt{\det M}} \\ &\quad \exp\left(\frac{1}{2} N^\top M^{-1} N - \frac{1}{2}\mathbf{r}^\top W \mathbf{r} - \frac{1}{2}\mathbf{s}^\top V \mathbf{s} - \frac{1}{2}\mathbb{E}[\boldsymbol{\beta}]^\top \mathbb{V}[\boldsymbol{\beta}]^{-1}\mathbb{E}[\boldsymbol{\beta}]\right), \end{aligned}$$

where

$$M = \mathbb{V}[\boldsymbol{\beta}]^{-1} + \begin{bmatrix} \sum_{a\in[m_a]} w_{ia} + \sum_{b\in[m_b]} v_{ia} & \sum_{a\in[m_a]} w_{ia}x_{ia} + \sum_{b\in[m_b]} v_{ib}y_{ib} \\ \sum_{a\in[m_a]} w_{ia}x_{ia} + \sum_{b\in[m_b]} v_{ia}y_{ia} & \sum_{a\in[m_a]} w_{ia}x_{ia}^2 + \sum_{b\in[m_b]} v_{ib}y_{ib}^2 \end{bmatrix} \qquad (5.25)$$

and

$$N = \mathbb{V}[\boldsymbol{\beta}]^{-1}\,\mathbb{E}[\boldsymbol{\beta}] + \begin{bmatrix} \sum_{a\in[m_a]} w_{ia}r_{ia} - \sum_{b\in[m_b]} v_{ib}s_{ib} \\ \sum_{a\in[m_a]} w_{ia}x_{ia}r_{ia} - \sum_{b\in[m_b]} v_{ib}y_{ib}s_{ib} \end{bmatrix}. \qquad (5.26)$$

Interested readers may refer to full derivation in the appendix.

Gaussian components Moreover, we may take into accounts of mean-shift of gene expression values. Suppose genes i and j are sampled from the same biological process. Then, we believe that there is the same classifier f that separates $\mathbf{x}_i^{(a)}$ versus $\mathbf{x}_i^{(b)}$, and $\mathbf{x}_j^{(a)}$ versus $\mathbf{x}_j^{(b)}$ for a discriminative pair (a, b); at the same time expression vectors $[\mathbf{x}_{ia}, \mathbf{x}_{ib}]$ and $[\mathbf{x}_{ja}, \mathbf{x}_{jb}]$ are similar to each other.

For each type t of samples, expression values x_{ij} take mean μ_t and precision r.

$$\mathcal{L}(\mathbf{x}_i^{(t)};\, \mu_t, r) = \prod_{j \in S_t} \mathcal{N}\left(x_{ij} \middle| \mu_t, r^{-1}\right) \tag{5.27}$$

Note the Gaussian distributions share the same precision for simplicity. For explicit representation, we also introduce z_i to indicate whether gene i was generated from this Gaussian components or not by 1 or 0. We assume conjugate prior to the precision r,

$$P(r|c_0, d_0) = \frac{(d_0)^{c_0}}{\Gamma(c_0)} r^{c_0 - 1} e^{-d_0 r}, \tag{5.28}$$

and another conjugate prior on top of all μ_t,

$$P(\mu_t|\mu_0, s, r) = \mathcal{N}\left(\mu_t \middle| \mu_0, (sr)^{-1}\right). \tag{5.29}$$

For posterior inference of $P(\mu_t|X)$ and $P(r|X)$, we used variational inference on the Gaussian components for consistency.

$$Q(\mu_t|\cdot) = \mathcal{N}\left(\mu_t \middle| \hat{\mu}_t, \gamma_t^{-1}\right) \tag{5.30}$$

where

$$\hat{\mu}_t = \frac{\mathbb{E}\left[\sum_{i \in [n]} \sum_{j \in S_t} z_i x_{ij}\right] + s\mu_0}{\mathbb{E}\left[\sum_{i \in [n]} z_i m_t\right] + s}, \quad \gamma_t = \mathbb{E}[r] \left(\mathbb{E}\left[\sum_{i \in [n]} m_t\right] + s\right).$$

and

$$Q(r|\cdot) = \mathsf{Gam}\left(r \middle| \hat{c}, \hat{d}\right) \tag{5.31}$$

where

$$\begin{aligned}\hat{c} &= c_0 + T/2 + \mathbb{E}\left[\sum_{i\in[n]} z_i \sum_t m_t\right]/2, \\ \hat{d} &= d_0 + \sum_{t\in[T]} \mathbb{E}\left[\sum_{i\in[n]} z_i \sum_{j\in S_t} (x_{ij}-\mu_t)^2 + s(\mu_t-\mu_0)^2\right].\end{aligned}$$

The Gaussian distribution adds a little more steps to the MED algorithm (Alg. 9). We add updates of $Q(\mu_t)$ and $Q(r)$, yielding Alg. 10. In the mixture setting, we multiply additional contribution to the MED (Eq. 5.24).

$$\begin{aligned}Q(\mathbf{x}_i|\cdot) \;\propto\; & \prod_{a<b} Q(\mathbf{x}_i^{(a)}, \mathbf{x}_i^{(b)}|\cdot) \\ & \int dr\, \mathsf{Gam}\left(r\middle|\hat{c},\hat{d}\right) \prod_{t\in[T]} \int d\mu_t\, \mathcal{N}\left(x_{ij}\middle|\mu_t, r^{-1}\right) \mathcal{N}\left(\mu_t\middle|\hat{\mu}_t, \gamma_t^{-1}\right).\end{aligned} \tag{5.32}$$

We omit complete evaluation of the second term for brevity, but interested readers may refer to the appendix.

Overall inference algorithm The MED-dpm has two variants, with or without Gaussian components. Overall algorithm is more or less the same (Alg. 11).

5.4 Neural stem cell differentiation

Gene sets / modules Gene sets are obtained from two sources. First are curated lists of genes, collected and annotated by domain experts, obtained from MSigDB [109]. Second are modules identified by automated community detection algorithms we have developed based on Bayesian methods and applied to biological interaction networks [140,142]. We implemented an improved version incorporating degree correction [89] because it performed better than other meth-

Alg 10 MED-Gauss

repeat
  repeat
    for $a < b$ **do**
      find optimal $\boldsymbol{\beta}^{(\hat{a},b)}$ by Eq. 5.18
      update the distribution $Q(\boldsymbol{\beta}^{(a,b)})$ by Eq. 5.21
      update the auxiliary $\mathbb{E}\left[(1/\tau^{(a,b)})^2\right]$ by Eq. 5.22
    end for
  until convergence of τ
  repeat
    update $Q(\mu_t)$ by Eq. 5.30
    update $Q(r)$ by Eq. 5.31
  until convergence of μ, r
  repeat
    estimate new penalty λ by Eq. 5.22
    reevaluate $\mathbb{E}\left[\tau^2\right]$ by Eq. 5.23
  until convergence of λ
until convergence of overall

Alg 11 MED-dpm

for $i \in [n]$ **do**
  uniform sampling of z_i from $[K]$
end for
repeat
  probabilistically assign data points to each k'th model
  inference of models by Alg. 9 or Alg. 10
  for $i \in [n]$ **do**
    collapsed Gibbs sampling of z_i to estimate $Q(z_i)$
    (Eq. 5.24 or Eq. 5.32)
  end for
until convergence of overall algorithm

ods (see Chapter 3). We constructed a physical interaction network of 14,995 proteins and 140,006 interactions from BioGRID 3.1.94 [174], which are edges labeled "physical." We also constructed a co-reaction network using current version of Reactome network database [32] [1], which contains 4,527 genes and 87,947 interactions, which includes pairs labeled "reaction" or "neighbouring_reaction" but not "direct complex" or "indirect complex" to avoid overlap with BioGRID. We resolved 173 physical modules and 144 co-reaction modules.

Neural stem cell differentiation Lesch-Nyhan Disease (LND) is a neurological disorder caused by mutations in HPRT (hypoxanthine guanine phosphoribosyltransferase), a purine biosynthesis gene [118]. While the overall disease mechanism remains puzzling, the mutation is thought to lead to defects in growth and differentiation of dopaminergic neurons [85]. We used time-series expression measurements from RNA-Seq data (GSE42662) for differentiation of spherical neural masses into dopaminergic neurons [86]. Samples were taken at 0, 1, 2, 3, 4, 6, 8, 10, 12, and 14 days, generated by human differentiation protocols applied to mouse embryonic stem cells for control vs. HPRT knockdown [28]. Three phases were observed: (i) neuronal induction, days 0–4; (ii) dopaminergic neuron induction, days 4–8; (3) dopaminergic neuron maturation, days 8–14.

Short reads were mapped to NCBI mm9 transcripts using `tophat` [185] (mapping rate $\sim$ 98%), counted by `htseq-count` [2] and normalized by `DESeq` taking into account gene length [4]. Mouse genes were then mapped to human genes `biomaRt` [203] for analysis with human gene sets.

[1] downloaded from `http://www.reactome.org/download/current/homo_sapiens.interactions.txt.gz` on Aug 4 2013

[2] `http://www-huber.embl.de/users/anders/HTSeq/doc/count.html`

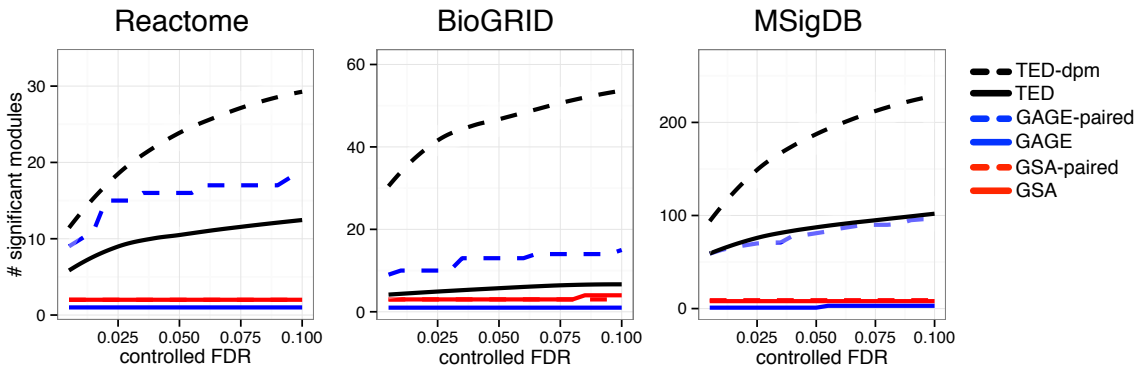

Figure 5.2: Significant modules identified for Lesch-Nyhan mouse model. The number of unique modules is shown for controlled FDR up to 0.1 (estimated 10% false discoveries).

Neural stem cell differentiation modules We then compared the number of modules identified by our discriminative methods, TED and TED-dpm, and existing methods (Fig. 5.2). With TED-dpm, we counted each original gene set at most once regardless of how many subsets it was partitioned into. TED-dpm identifies a greater number of significant modules than other methods. GAGE-paired performs as well as the base TED method for literature pathways from MSigDB, and better than TED for modules predicted by community detection algorithms for Reactome and BioGRID, but never as well as TED-dpm. The unpaired GAGE based method and paired and unpaired variants of GSA perform poorly on this data set.

We then investigated the modules from Reactome and BioGRID identified by TED (Fig. 5.3) and TED-dpm (Fig. 5.4) using a stringent FDR cutoff of 0.01 (1% false discoveries). Modules were annotated by comparison with MSigDB canonical gene sets requiring a hypergeometric FDR of 0.01. Applying TED-dpm identifies significant sub-sets purified for expression change and, by comparison with curated pathways, for known biological function (Fig. 5.4). Throughout

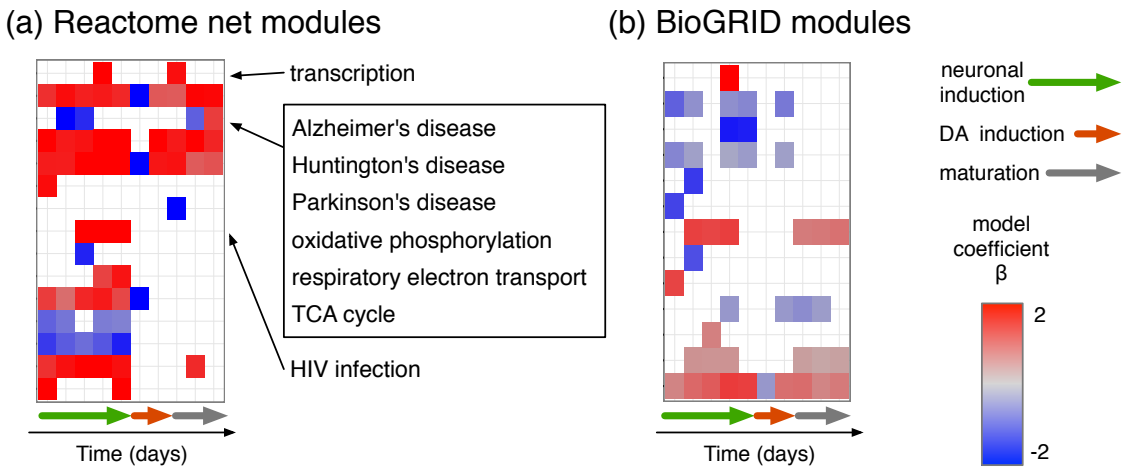

Figure 5.3: Differentially regulated network modules found by TED. TED identifies 15 Reactome modules and 13 BioGRID modules from a mouse model for dopaminergic (DA) neuron development. Each module has an associated β_t and p-value at each time point, and colors indicated p-values that are significant at FDR = 0.01. Up-regulation in control corresponds to $\beta_t > 0$, red; down-regulation in control is $\beta_t < 0$, blue.

the time course, modules related to cell cycle, nucleotide and downstream lipid metabolism were up-regulated in control samples; the corresponding genes are likely less active in the disease state. Modules corresponding to GPCR signaling (Reactome) and portions of TGF-β signaling (BioGRID) were up-regulated in the HPRT knockdown cells. Some modules show a change in direction of control-vs-knockdown gene expression, for example RNA processing in the Reactome modules (Fig. 5.4a), and biologically uncharacterized modules from BioGRID (Fig. 5.4). These modules may have been difficult to identify using standard methods if expression differences over a time course cancel.

Glycosaminoglycan complex The glycosaminoglycan module from Reactome illustrates the ability of TED-dpm to identify sub-networks purified for gene expression and, potentially, biological function (Fig. 5.5). TED-dpm identifies a

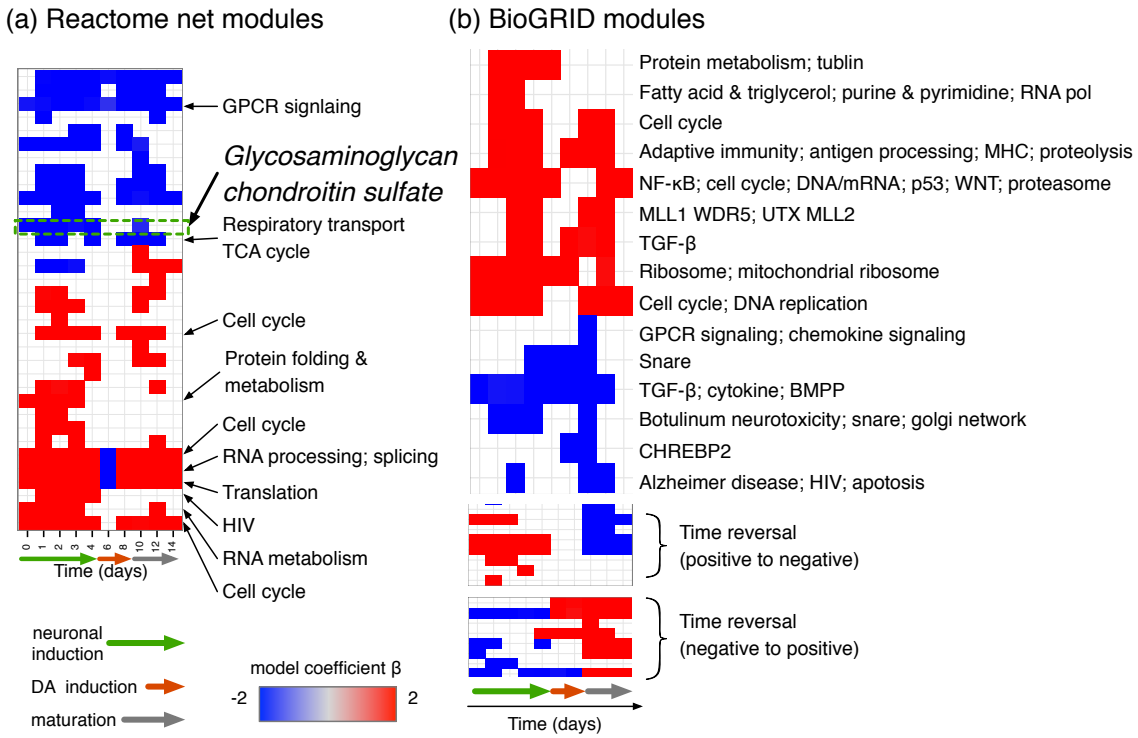

Figure 5.4: Differentially regulated network modules found by TED-dpm. Colored squares indicate significant discrimination of controls-vs-knockdown for the indicated module and time point at FDR = 0.01. Names indicate overlap with canonical gene sets from MSigDB at hypergeometric FDR 0.01. Red and blue indicate up-regulation and down-regulation of control-vs-knockdown.

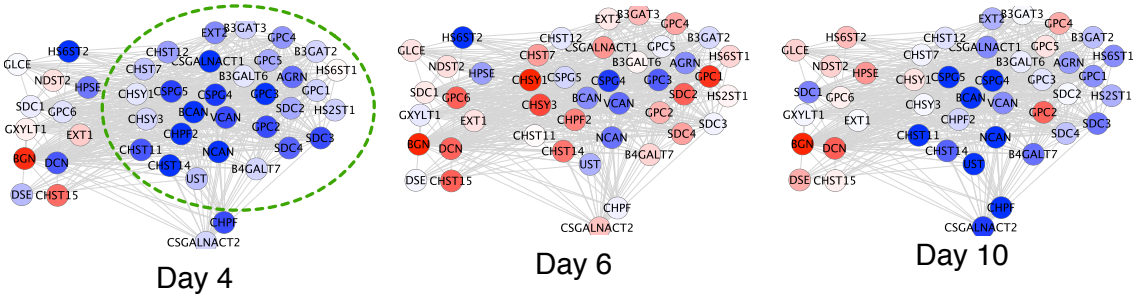

Figure 5.5: Transcriptomic dynamics of "Glycosaminoglycan" module in Reactome network. Green dashed circle encloses parts included by TED-dpm. Nodes and edges represent genes and interactions of Reactome network (co-reaction). Genes were colored by relative expression level, red for strong $\log_2$(control/KD) ratio and blue for strong $\log_2$(KD/control) ratio.

sub-complex (green dashed circle) that contains glycan and chondroitin enzymes: B4GALT (β-1,4-galactosyltransferase), B3GAT (β-1,3-glucuronyltransferase), CHST (carbohydrate sulfotransferase), CHSY (chondroitin synthase), and CHPF (chondroitin polymerizing factor). In the HPRT knockdown, this core complex was initially up-regulated during neuronal induction phase until day 4, but gave a less discriminative signal afterwards (Fig. 5.4a; Fig. 5.5). These findings from discriminative learning are consistent with hypotheses that neural stem cells can be identified specific glycan markers, with neuronal lineages regulated and marked by post-translational modifications of glycans attached to membrane [104]. The mRNA levels of many glycan enzymes strongly correlate with mouse embryonic stem cell fate [129], suggesting that Lesch-Nyhan disease may arise from improper cell fate determination. In some cases, abnormal expression B4GALT family genes have been shown to promote multi-drug resistance in leukemia cells by regulating hedgehog signaling [204].

5.5 Differential responses to four cytokines

We obtained four types of time-series microarrays measured on Human Umbilical Vein Endothelial (HUVEC) cells with four different treatments of cytokines: IL-1-treated (GSE973) [119], TNF-α-treated (GSE 9055) [189], VEGF-treated (GSE10778) and EGF-treated (GSE10778) [168]. These data sets provide genomic profile of gene activities in response to different external stimuli. We were interested in looking for differential regulatory modules that may reveal mechanism of cell lineage.

Microarray data pre-processing From each array we removed genes in the lower 20% quantile to avoid technical artifacts (e.g., log 0), and calculate log2 ratio compared to the control (0 minute expression values). We normalized to have expressions scale similarly across data sets. First we converted the log2-ratio values x_{ij} to z-scores $z_{ij} = (x_{ij} - m_j)/s_j$, where m_j and s_j are the sample mean and standard deviation of array j. We then turned the z-score to truncated p-values, $p_{ij} = \max\{10^{-5}, 2F(|z_{ij}|)\}$, where $F(x)$ is $1 - \Phi(x)$ and $\Phi(x)$ is cumulative distribution function of the standard Normal distribution. Truncation is crucial since it prevents from numerical underflow and overshoot of outliers. Finally we converted pij back to scaled with a proper sign, $x_{ij} = \mathsf{sign}(x_{ij})F^{-1}(p_{ij})$. We managed to apply other pooling methods, available in the Bioconductor, such as the virtual array [69], but none of them worked properly. Potentially an improved pooling method would improve accuracy of downstream analysis, but we expect results would remain qualitatively and quantitatively invariant.

Sources of gene sets We used the same gene sets / modules previously identified for the Lesch-Nyhan disease study: network modules identified in BioGRID [174] and Reactome networks [32], and MSigDB canonical pathways [109].

Effect of Gaussian components We demonstrate that modeling mean-shifts made by Gaussian distributions help distinguish subtle difference within modules. In the examples of Reactome modules we found the additional Gaussian components make results more clean and interpretable (Fig. 5.6). Partitioned subsets are unimodal in both sense of discriminative learning and distribution. In this example, at the same threshold (FDR less than 5%) we were able to find

more modules (A), and the modules are more distinctive in overall distribution (B). It helps improve statistical power while controlling type-I error more effectively. In the follow-up analysis we show the results of MED-dpm that includes Gaussian components.

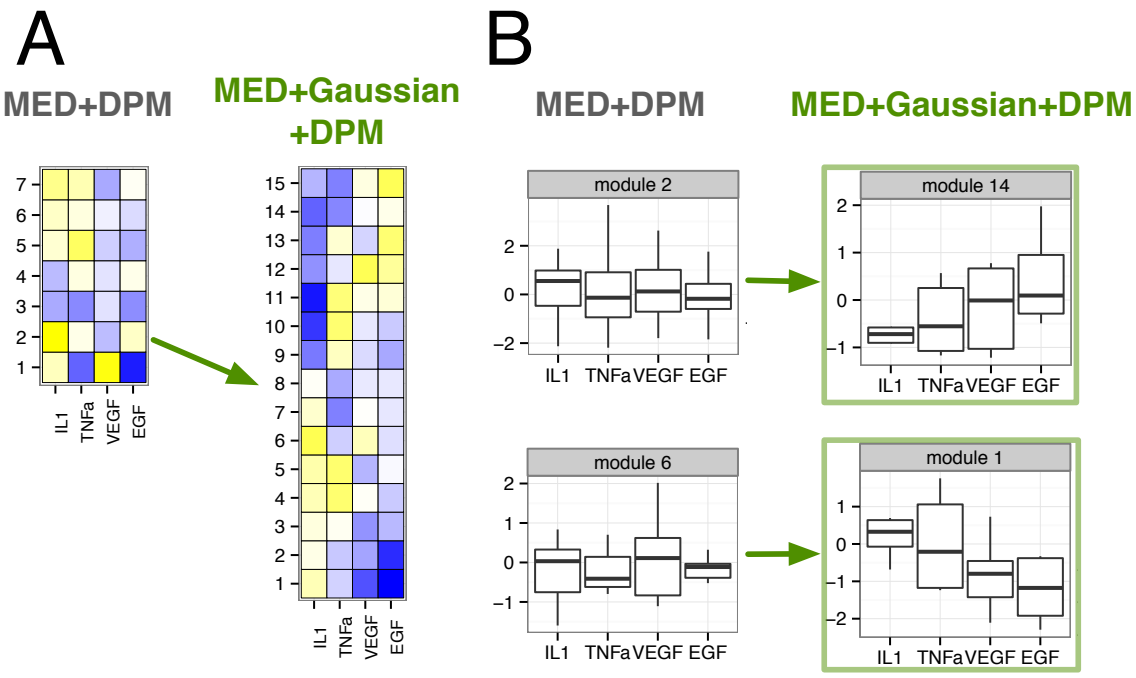

Figure 5.6: Effects of Gaussian components. (A) Median expressions of differentially regulated modules identified by MED-dpm with or without Gaussian components. (B) Exemplary box-plots summarize the distribution of gene expressions in significant modules identified by both methods.

Differentially regulated Reactome modules Of hundreds of Reactome modules, we focus on to 15 differentially regulated modules significant in at least one pairwise test (Eq. 5.14) and one combined test (Eq. 5.15) at FDR less than 5%, and all of the network modules well matched with known canonical pathways in MSigDB at FDR less than 5% (Fig. 5.7). We ordered these modules (rows) by hi-

erarchical clustering based on the Euclidean distance between modules (`hclust` in R/Splus). The bottom 8 modules (1-8) are generally up-regulated in the IL-1-treated, down-regulated in the EGF-treated samples, whereas the top 8 modules are strongly down-regulated in the IL-1-treated.

The result is actually self-proving biological relevance. IL-1 strongly triggers the modules #1, 4, and 5 to 7, but represses the modules 9 to 10 (Fig. 5.7B). Of the activated modules, the module #5 is directly related to downstream signaling pathways of interleukin; and GPCR downstream pathways (module #6) were known to respond to IL-1β in experimental studies [145]. Moreover, we verify that EGF-treatment induces the cell-cycle module (module #13).

We find the profiles of TNF-α treated cells are generally similar to the IL-1-treated that the others, and the VEGF and EGF-treated cells are similar to each other (Fig. 5.7A). The coupling of the TNF-α and IL-1 treatments coincides with the fact that TNF-α induces expression of IL-1 [184]. However, the strong up-regulations of transcription (module #11) and translation (module #10) are only specific to the TNF-α-treated cells. These modules indeed characterize difference between the IL-1 and TNF-α (the green dots of modules #10-11 in Fig. 5.7C).

Particularly we are interested in the GPCR signaling modules (#6, 9 and 12) because these modules were initially co-clustered in network structure, but divided into separate modules by the MED-dpm (Fig. 5.8A). More interestingly, Although these modules are tightly connected in the Reactome network, suggesting strong evidence of co-regulation (Fig. 5.8B), different combination of directions in regulation sufficiently discriminate all four conditions. For instance we may code each treatment by three binary digits, $b_6 b_9 b_{12}$, where $b_j = 1$ for up-regulation of module j, and 0 for down-regulation (Table. 5.1).

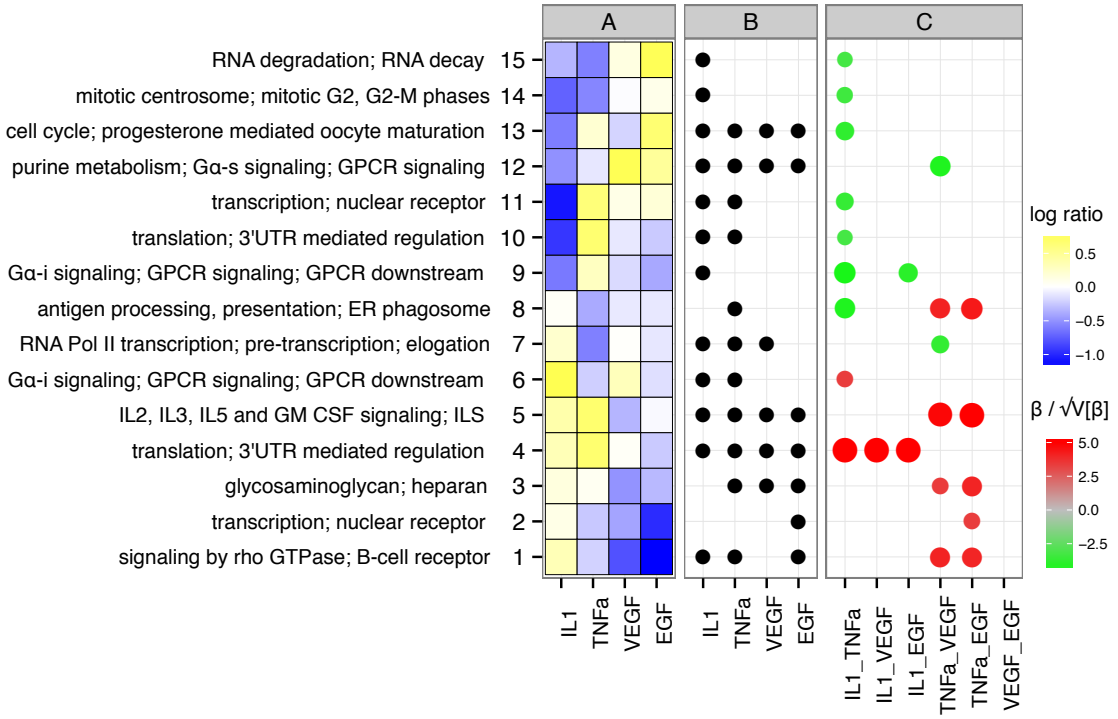

Figure 5.7: Differentially regulated Reactome modules. (A) Median gene expressions of the modules changing from blue to yellow. (B) Dots indicate genome-wide significant conditions per module in the combined test (Eq. 5.15) at FDR$<$ 0.05. (C) Dots indicate genome-wide significant pairs of conditions per module (Eq. 5.14) at FDR$<$ 0.05; colors denote the direction of classification rule, where for a pair A_B, the red indicates A$>$B and the green indicates the opposite, A$<$B. For each module (row) we annotate overlapping canonical pathways determined by hypergeometric test at FDR$<$ 0.05.

code $b_6 b_9 b_{12}$	treatment
100	IL-1
010	TNF-α
101	VEGF
001	EGF

Table 5.1: The regulatory code of GPCR modules. 1: up-regulation; 0: down-regulation after the treatment.

Differentially regulated BioGRID modules In the BioGRID physical network we only discovered 6 network modules are strongly discriminative by the same

threshold applied to the Reactome network, significance in at least one pairwise test (Eq. 5.14) and one combined test (Eq. 5.15) at FDR$<$ 0.05. Of 6 significant modules, two modules, #3 and #6, overlap with the known canonical pathways (Fig. 5.9A). The genes in the module #3 are strongly connected, and well separate IL-1 and TNF-α from VEGF and EGF (Fig. 5.9B). Functionally genes correspond to the NF-κB atypical pathway and up-regulated by the treatment of TNF-α and IL-1. However, there is a large amount of variation within TNF-α samples, and follow-up time-series analysis of this module would reveal more detailed mechanisms.

5.6 Biological impact

A key motivation behind this chapter was very practical. From static or dynamic network clustering we identify network modules. Especially the proposed methods in this research are capable of obtaining high resolution of modularization, meaning a large number of modules. Network clustering helps zooming into a specific set of genes, but the problem is that we do not know where to zoom-in. Networks are generally considered context-free, not all modules are of interest to all contexts. Therefore we borrow additional information focus on a specific context that was gene expression data sets in this case.

Modules identified for Lesch-Nyhan disease may lead to new hypotheses about disease mechanism, which is thought to relate to neuronal differentiation but is largely unknown. We also discovered modules reverse the direction of expressions, which would remain uncovered unless we take into accounts of temporal axis. Discriminative modules identified for the cytokine study help under-

stand downstream regulations affected by drug treatments. Succinct regulatory codes of GPCR pathways provide testable working hypothesis for small-scale validation experiments (Table. 5.1).

We envision, in future directions, that given gene sets / modules we may use a large compendium of expression databases such as Gene Expression Omnibus [3] to provide complete and systematic views of biological systems. For instance we may interrogate tissue-type variability in evolutionary context [170]; or, we could add genotype data sets to the discriminative models [111].

5.7 Technical impact

TED to identify gene sets that discriminate between biological states, and TED-dpm improves upon TED by using a Dirichlet process mixture to purify sub-sets for consistent expression. The base method TED is competitive with the best-performing enrichment-based methods. TED-dpm performs better than competing methods based on enrichment for simulated data, and identifies a greater number of significant modules for a biological data set generated from a mouse model for Lesch-Nyhan disease.

Admittedly we made a strong assumption to the MED model, that time-course expressions are nearly identical within each cytokine treatment. We may consider imputing unobserved time points and match data sets under the assumption of smoothness (e.g., kernelization [172]). However, a much better solution is obviously that we need better design of studies.

Part of the gain in performance arises from the Dirichlet process mixture,

[3] `www.ncbi.nlm.nih.gov/geo`

which could potentially be used to improve the performance of enrichment-based methods as well. We were able to subtype smaller coherent subsets. from a large multi-modal data set. This implicates a different direction of research. For instance we may consider a Latent Dirichlet Allocation of expression topics [23], where the notion of topic could be TED / MED model.

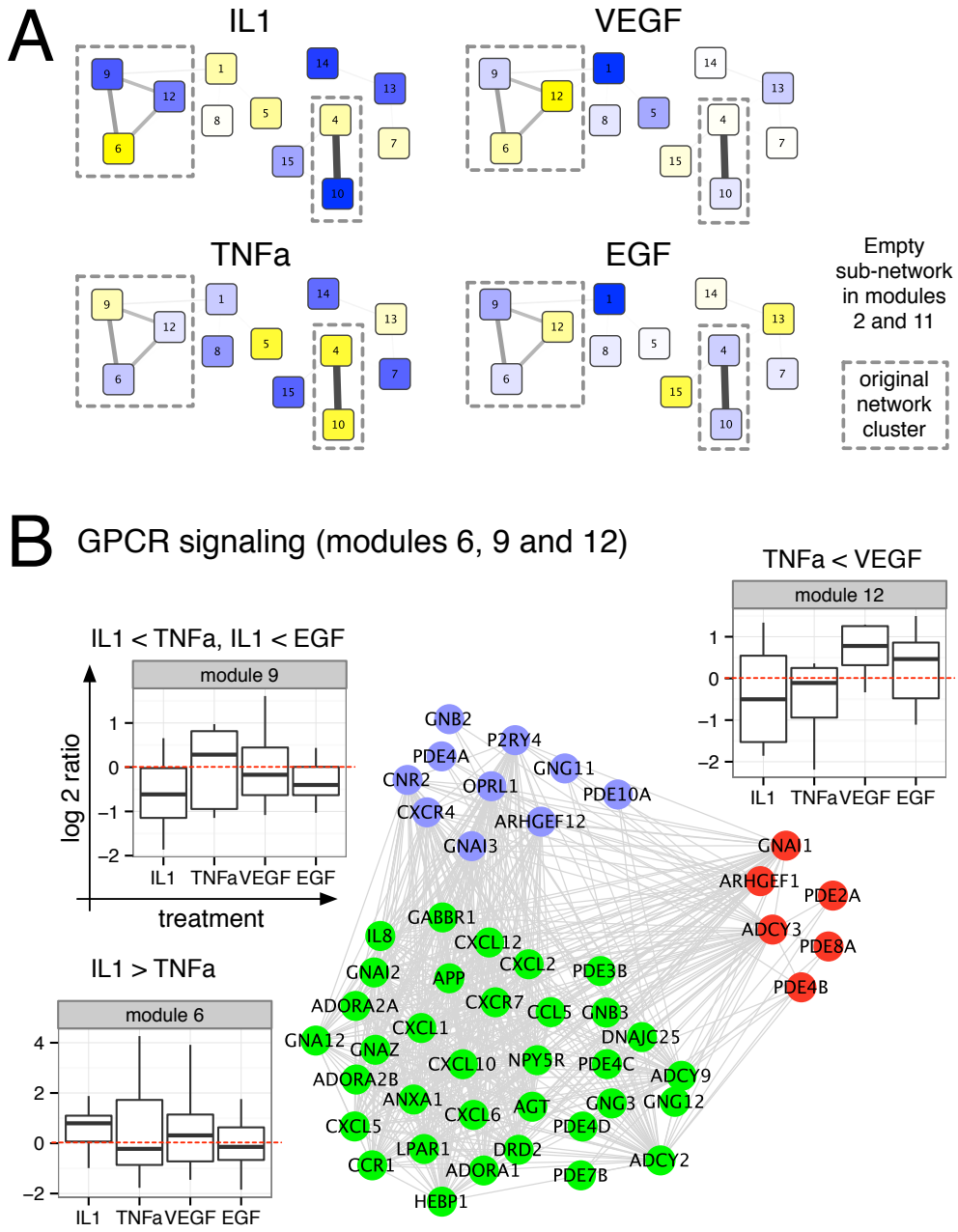

Figure 5.8: Differentially regulated Reactome modules. (A) *Modular networks under different conditions.* Vertices correspond to modules and edge widths scale proportional to the probability of interaction between modules. Vertices are colored by median gene expressions of the modules changing from blue to yellow. (B) *Zoomed-in view of modules #6, #9 and #12.* The modules are colored differently: #6 with green; #9 with light blue; #12 with red. *The box plots* show distribution of gene expression in response to different cytokines. Network diagrams visualize protein-protein interactions occurring within the module. The network in the module #6 consists multiple connected components; here, we show the largest component.

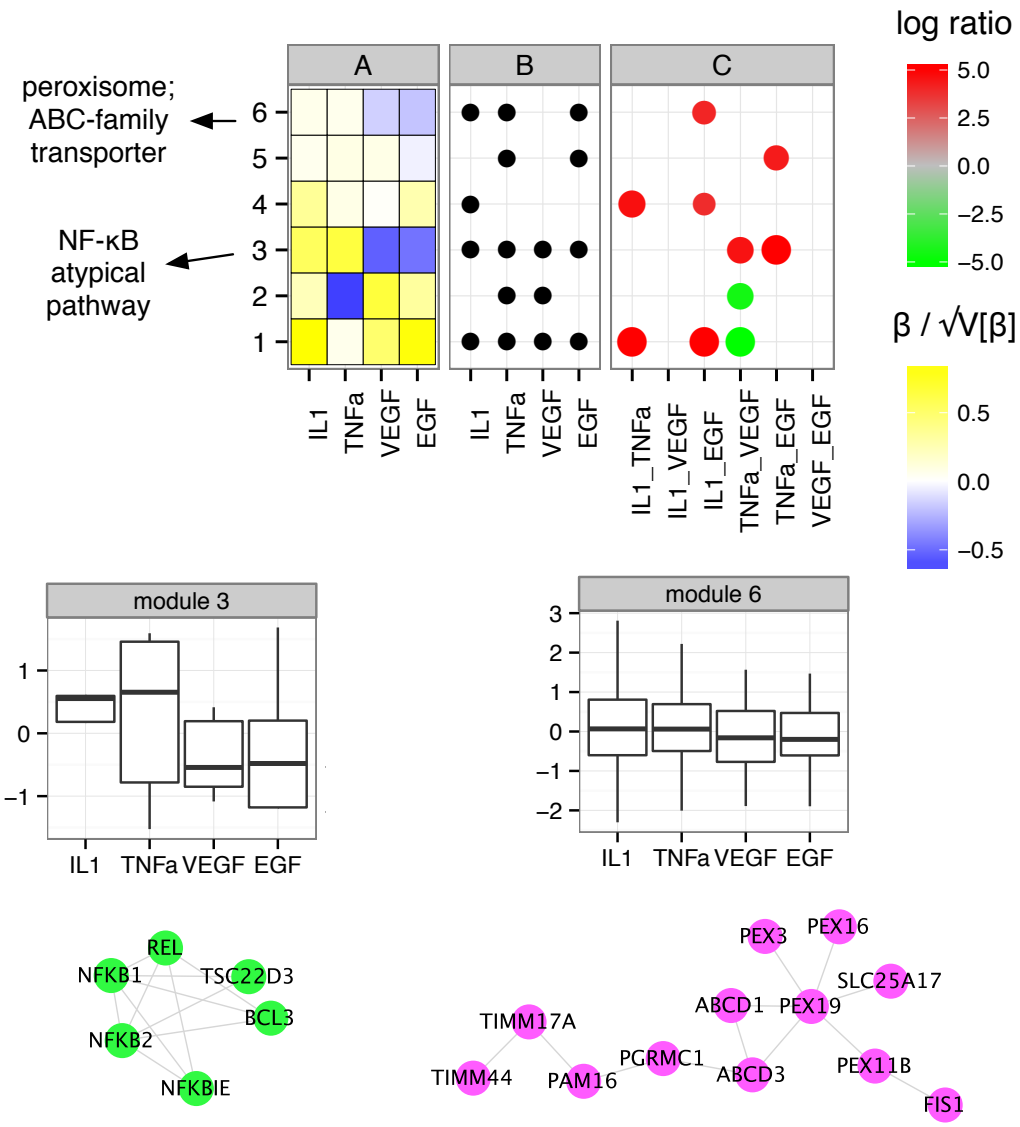

Figure 5.9: Differentially regulated BioGRID modules. (A) Median gene expressions of the modules changing from blue to yellow. (B) Dots indicate genome-wide significant conditions per module in the combined test (Eq. 5.15) at FDR$<$ 0.05. (C) Dots indicate genome-wide significant pairs of conditions per module (Eq. 5.14) at FDR$<$ 0.05; colors denote the direction of classification rule, where for a pair A_B, the red indicates A$>$B and the green indicates the opposite, A$<$B. (module #) The box plot shows distribution of gene expression under different treatments. Network diagrams visualize protein-protein interactions occurring within the modules. The network in the module 6 consists multiple connected components; here, we show the largest component.

Chapter 6
Conclusion

Implications Over the course of research, we analyzed a wide spectrum of biological networks, especially physical interaction networks, by multiple algorithms. From careful experiments and interpretation of the results we provide best possible answers to the fundamental / practical questions about biological networks (Chapter. 1) and these answers are summarized as the following thesis statements:

1. Biological networks are best explained by a model that fits hierarchical block structures. We also propose to revise a common notion of block / cluster / module / community in the network. At least in biological networks most predominant form was a hub-and-spoke subnetworks, rather than cliques.

2. Network modules generally remain intact over the temporal cycles / biological process. Genes / proteins in the same modules appear and disappear almost synchronously, and most members are usually bounded to a simple complex. Complete rearrangement of clustering would be very unusual. However in each module there are a handful of local modulators, dynamically switching on and off, and regulate other members.

3. However, even tightly connected network modules (subnetworks) may break into smaller subunits while differently responding to external and internal biological contexts. Therefore, researchers may need consider partitions of predefined modules and gene sets, adaptively to the given experimental conditions.

Future directions Most state-of-the-art network modeling builds upon the stochastic block model [75], which bases on the stochastic equivalence of vertices (Def. 6). Even for the recently developed link community models [2, 10] a similar equivalence relation must be assumed. However, unless we observe dynamic edges this framework is largely limited to static networks, and prevents from brining in diverse and rich contextual information of "omics" data sets. Along this direction there were several attempts in social network analysis [11, 93]; yet, approaches are rather *ad hoc*, problem-specific and just builds upon well-established block models. In systems biology problems we would need more biologically relevant definition of equivalence considering genomic / transcriptomic aspects.

Computational approaches are usually considered as "data analysis" steps after generation of large-scale data sets. However, we know biological networks are incomplete [201], and any models trained on the incomplete data sets are also incomplete. We need a better way to do science, maximizing the usage of scalable computational methods, especially Bayesian inference algorithms, taking full advantages of big data, so that computational communities provide positive inputs to data generation process. Therefore method development should embrace active learning frameworks [127], and development of online recommendation and "never-ending learning" systems could be beneficial as well [26].

Chapter 7

Appendix: mathematical details of Chapter. 5

7.1 Justification of Bayesian (fused) Lasso

We simply reiterate known results [100]. We can recover the Fused Lasso [181] by integrating out auxiliary τ and κ,

$$P(\beta_j) = \int_0^\infty \mathcal{N}\left(\beta_j \middle| 0, \tau_j^2\right) \mathsf{Exp}\left(\tau_j^2 \middle| \lambda^2/2\right) d\tau_j^2 = \frac{\lambda}{2} e^{-\lambda|\beta_j|}$$

and

$$P(\beta_j - \beta_{j+1}) = \int_0^\infty \mathcal{N}\left(\beta_j - \beta_{j+1} \middle| 0, \kappa_j^2\right) \mathsf{Exp}\left(\kappa_j^2 \middle| \gamma^2/2\right) d\kappa_j^2 = \frac{\gamma}{2} e^{-\gamma|\beta_j - \beta_{j+1}|}.$$

More explicitly,

$$\begin{aligned}
P(\beta) &= \int dv\, \mathcal{N}(\beta|0,v)\, \mathsf{Exp}\left(v \middle| \lambda^2/2\right) \\
&= \int_0^\infty dv \sqrt{\frac{1}{2\pi v}} \frac{\lambda^2}{2} \exp\left\{-\frac{\lambda^2}{2} v - \frac{\beta^2}{2} v^{-1}\right\} \\
&= \lambda^2 \sqrt{\frac{1}{2\pi}} \int_0^\infty dv\, v^{-1/2} \frac{1}{2} \exp\left\{-\frac{\lambda^2}{2} v - \frac{\beta^2}{2} v^{-1}\right\} \\
&= \lambda^2 \sqrt{\frac{1}{2\pi}} \int_0^\infty du\, \exp\left\{-\frac{\lambda^2}{2} u^2 - \frac{\beta^2}{2} u^{-2}\right\} \\
&= \lambda^2 \sqrt{\frac{1}{2\pi}} \sqrt{\frac{\pi}{2\lambda^2}} \exp\{-\lambda|\beta|\} \\
&= \frac{\lambda}{2} \exp\{-\lambda|\beta|\}.
\end{aligned}$$

Here we used the relation

$$\int_0^\infty \exp\{-\frac{1}{2}[a^2u^2 + b^2u^{-2}]\}du = \sqrt{\frac{\pi}{2a^2}}\exp\{-|a||b|\},$$

from the inverse Gaussian distribution. Rearranging,

$$\int_0^\infty \exp\{-\frac{1}{2}(a^2u^2 + b^2/u^2)\}du = \exp\{-|a||b|\}\frac{1}{2}\int_0^\infty \exp\{-\frac{1}{2u^2}(au^2 - b)^2\}du.$$

The second term is

$$\begin{aligned} C &= \frac{1}{2}\int_0^\infty s^{-1/2}\exp\{-\frac{1}{2s}(as-b)^2\}ds \\ &= \frac{1}{2}\int_0^\infty r^{-3/2}\exp\{-\frac{r}{2}(a/r-b)^2\}dr \\ &= \frac{1}{2}\int_0^\infty r^{-3/2}\exp\{-\frac{a^2}{2}\frac{(r-a/b)^2}{r(a/b)^2}\}dr \\ &= \sqrt{\frac{\pi}{2a^2}}, \end{aligned}$$

with the last equality from the Inverse-Gaussian p.d.f.

$$P(x|\mu,\sigma^2) = \sqrt{\frac{\sigma^2}{2\pi}}x^{-3/2}\exp\left\{-\frac{\sigma^2}{2\mu^2 x}(x-\mu)^2\right\}.$$

7.2 Derivation of τ and κ

We have used posterior probability

$$1/\tau_t^2 \sim \text{inv}\mathcal{N}\left(1/\tau_t^2\middle|\sqrt{\frac{\lambda^2}{\beta_t^2}},\lambda^2\right)$$

and

$$1/\kappa_t^2 \sim \text{inv}\mathcal{N}\left(1/\kappa_t^2\middle|\sqrt{\frac{\gamma^2}{(\beta_t-\beta_{t-1})^2}},\gamma^2\right)$$

when updating τ and κ (Eq. 5.9). We justify these explicitly and derive update equations. Let $v = \tau^2$. Given $Q(\beta)$, we want to estimate $Q(v)$. Rearrangement of equations gives

$$\begin{aligned} Q(v|\cdot) &\propto P(\beta|v)P(v|\lambda) \\ &\propto \frac{1}{\tau}\exp\left\{-\frac{1}{2}\left[\lambda^2 v + \mathbb{E}\left[\beta^2\right]/v\right]\right\}. \end{aligned}$$

We can characterize the distribution of $u \equiv 1/v$. By the transformation of random variable,

$$\begin{aligned} Q(u|\cdot) &\propto u^{-3/2} \exp\{-\frac{1}{2}(\sqrt{\mathbb{E}[\beta^2]}\sqrt{u} - \lambda/\sqrt{u})^2\} \\ &\propto u^{-3/2} \exp\{-\frac{\lambda^2(u - \lambda/\sqrt{\mathbb{E}[\beta^2]})^2}{2u\lambda^2/(\mathbb{E}[\beta^2])}\}. \end{aligned}$$

Therefore,

$$Q(u|\cdot) = \text{inv}\mathcal{N}\left(u \middle| \sqrt{\frac{\lambda^2}{\mathbb{E}[\beta^2]}}, \lambda^2\right).$$

The inverse Gaussian distribution can be rewritten in the exponential family form:

$$P(x|\mu, \xi) = h(x) \exp\left\{(a, b)\,(x, -1/x)^\top - A(a, b)\right\}$$

where $a = -\xi/2\mu^2$ and $b = -\xi/2$, and

$$A(a, b) = -2\sqrt{ab} + \frac{1}{2}\log(-2b), \quad h(x) = x^{-3/2}.$$

This allows us to easily calculate

$$\mathbb{E}[x] = \mu, \quad \mathbb{E}[1/x] = 1/\mu + 1/\xi.$$

Therefore we can characterize the expectations of v and v^{-1} by $Q(u)$ as

$$\mathbb{E}\left[1/\tau^2\right] = \mathbb{E}\left[v^{-1}\right] = \frac{\lambda}{\sqrt{\mathbb{E}[\beta^2]}}, \quad \mathbb{E}\left[\tau^2\right] = \mathbb{E}[v] = \frac{\sqrt{\mathbb{E}[\beta^2]}}{\lambda} + \frac{1}{\lambda^2},$$

which is equivalent to Eq. 5.9. We can derive update equations of κ similarly.

7.3 Locally Collapsed Variational Inference of TED

To circumvent non-conjugate relations, we approximate the likelihood by second order Taylor expansion,

$$\log \mathcal{L}(\mathbf{x}, \mathbf{y}; \boldsymbol{\beta}) \approx \sum_{t=1}^{T} \left[f(\hat{\beta}_t) + f'(\hat{\beta}_t)(\beta_t - \hat{\beta}_t) + \frac{1}{2} f''(\hat{\beta}_t)(\beta_t - \hat{\beta}_t)^2 \right], \tag{7.1}$$

where $f(\beta_t) = -\log(1 + e^{-\beta_t x_t}) - \log(1 + e^{\beta_t y_t})$. Then, posterior probability of $(\mathbf{x}_i, \mathbf{y}_i)$ assignment to model k is straightforward.

First let us see the following relation:

$$\begin{aligned} S(x) &= \exp\left(f'(\mu)x + \frac{1}{2}f''(\mu)(x-\mu)^2 - \frac{1}{2}\lambda(x-\mu)^2\right) \\ &= \exp\left(-\frac{\tilde{\lambda}}{2}(x-\mu)^2 + f'(\mu)x\right) \\ &= \exp\left(-\frac{\tilde{\lambda}}{2}\left[x - 2\tilde{\mu}x + \tilde{\mu}^2\right] - \frac{\tilde{\lambda}}{2}\mu^2 + \frac{\tilde{\lambda}}{2}\tilde{\mu}^2\right) \\ &= \exp\left(-\frac{\tilde{\lambda}}{2}(x-\tilde{\mu})^2 - \frac{\tilde{\lambda}}{2}\mu^2 + \frac{\tilde{\lambda}}{2}\tilde{\mu}^2\right) \end{aligned}$$

where we simplified $\tilde{\lambda} \equiv (\lambda - f''(\mu))$ and $\tilde{\mu} \equiv (\mu + f'(\mu)/\tilde{\lambda})$. Then,

$$\int S(x)\,dx = \exp\left(-\frac{\tilde{\lambda}}{2}\mu^2 + \frac{\tilde{\lambda}}{2}\tilde{\mu}^2\right)\left(\frac{\tilde{\lambda}}{2\pi}\right)^{-1/2}.$$

Using this, we can easily evaluate our update equation:

$$\begin{aligned} \mathbb{E}\left[e^{\sum_t f(\beta_t)}\right] &= \prod_t \int_{-\infty}^{\infty} \exp(f(\beta))\,\mathcal{N}\left(\beta\middle|\hat{\beta}_t, \lambda_t^{-1}\right)\,d\beta \\ &\approx \prod_t \int_{-\infty}^{\infty} \exp\left(f(\hat{\beta}_t) + f'(\hat{\beta}_t)(\beta-\hat{\beta}_t) + \frac{1}{2}f''(\hat{\beta}_t)(\beta-\hat{\beta}_t)^2\right) \\ &\quad \mathcal{N}\left(\beta\middle|\hat{\beta}_t, \lambda_t^{-1}\right)\,d\beta \\ &= \prod_t \exp\left(f(\hat{\beta}_t) - f'(\hat{\beta}_t)\hat{\beta}_t\right)\exp\left(-\frac{\tilde{\lambda}_t}{2}\mu^2 + \frac{\tilde{\lambda}_t}{2}\tilde{\mu}^2\right)\left(\frac{\lambda_t}{\tilde{\lambda}_t}\right)^{1/2} \\ &= \prod_t \exp\left(f(\hat{\beta}_t) + \frac{1}{2}\frac{f'(\hat{\beta}_t)^2}{\lambda_t - f''(\hat{\beta}_t)}\right)\left(\frac{\lambda_t}{\lambda_t - f''(\hat{\beta}_t)}\right)^{1/2}. \end{aligned}$$

7.4 Locally collapsed variational inference of MED

Let us simply $\exp\{g_i(\mathbf{x}_i, \mathbf{y}_i; \boldsymbol{\beta})\}$ (Eq. 5.17) by matrix notation. Let

$$X = \begin{bmatrix} 1 & x_{i1} \\ \vdots & \\ 1 & x_{im_a} \end{bmatrix}, \quad Y = \begin{bmatrix} -1 & -y_{i1} \\ \vdots & \\ -1 & -y_{im_b} \end{bmatrix},$$

$$\boldsymbol{\beta} = \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix}, \quad \mathbf{r} = \begin{bmatrix} r_{i1} \\ \vdots \\ r_{im_a} \end{bmatrix}, \quad \mathbf{s} = \begin{bmatrix} s_{i1} \\ \vdots \\ s_{im_b} \end{bmatrix}$$

and

$$W = \begin{bmatrix} w_{i1} & 0 & \cdots & 0 \\ & \vdots & & \\ 0 & 0 & \cdots & w_{im_a} \end{bmatrix}, \quad V = \begin{bmatrix} v_{i1} & 0 & \cdots & 0 \\ & \vdots & & \\ 0 & 0 & \cdots & v_{im_b} \end{bmatrix}.$$

Let

$$M = \mathbb{V}[\boldsymbol{\beta}]^{-1} + X^\top W X + Y^\top V Y, \quad N = \mathbb{V}[\boldsymbol{\beta}]^{-1}\mathbb{E}[\boldsymbol{\beta}] + X^\top W \mathbf{r} + Y^\top V \mathbf{s},$$

which are equivalent to Eq. 5.25 and Eq. 5.26. We have

$$\log P(\mathbf{x}_i, \mathbf{y}_i, \boldsymbol{\beta}) \approx f(\boldsymbol{\beta}) + \text{const.}$$

where

$$\begin{aligned} f(\boldsymbol{\beta}) &= g_i(\boldsymbol{\beta}) - \frac{1}{2}(\boldsymbol{\beta} - \mathbb{E}[\boldsymbol{\beta}])^\top \mathbb{V}[\boldsymbol{\beta}]^{-1}(\boldsymbol{\beta} - \mathbb{E}[\boldsymbol{\beta}]) + \log\det \mathbb{V}[\boldsymbol{\beta}]^{-1} \\ &= -\frac{1}{2}\left[\boldsymbol{\beta}^\top M \boldsymbol{\beta}^\top - 2N^\top \boldsymbol{\beta}\right] \\ &\quad -\frac{1}{2}\mathbf{r}^\top W \mathbf{r} - \frac{1}{2}\mathbf{s}^\top V \mathbf{s} - \frac{1}{2}\mathbb{E}[\boldsymbol{\beta}]^\top \mathbb{V}[\boldsymbol{\beta}]^{-1}\mathbb{E}[\boldsymbol{\beta}] + \frac{1}{2}\log\det \mathbb{V}[\boldsymbol{\beta}]^{-1} \\ &= -\frac{1}{2}(\boldsymbol{\beta} - M^{-1}N)^\top M(\boldsymbol{\beta} - M^{-1}N) + \frac{1}{2}N^\top M^{-1} N \\ &\quad -\frac{1}{2}\mathbf{r}^\top W \mathbf{r} - \frac{1}{2}\mathbf{s}^\top V \mathbf{s} - \frac{1}{2}\mathbb{E}[\boldsymbol{\beta}]^\top \mathbb{V}[\boldsymbol{\beta}]^{-1}\mathbb{E}[\boldsymbol{\beta}] + \frac{1}{2}\log\det \mathbb{V}[\boldsymbol{\beta}]^{-1}. \end{aligned}$$

From the fact derived from the multivariate Gaussian

$$2\pi(\det M)^{-1/2} = \int d\boldsymbol{\beta} \exp\left(-\frac{1}{2}(\boldsymbol{\beta} - M^{-1}N)^\top M(\boldsymbol{\beta} - M^{-1}N)\right),$$

we resolve

$$\begin{aligned} \mathbb{E}\left[e^{f(\boldsymbol{\beta})}\right] \propto & \frac{1}{\sqrt{\det \mathbb{V}[\boldsymbol{\beta}]}\sqrt{\det M}} \exp\left(\frac{1}{2}N^\top M^{-1} N - \frac{1}{2}\mathbf{r}^\top W \mathbf{r} - \frac{1}{2}\mathbf{s}^\top V \mathbf{s}\right. \\ & \left. -\frac{1}{2}\mathbb{E}[\boldsymbol{\beta}]^\top \mathbb{V}[\boldsymbol{\beta}]^{-1}\mathbb{E}[\boldsymbol{\beta}]\right). \end{aligned}$$

And we approximate $Q(\mathbf{x}_i, \mathbf{y}_i|\cdot) \propto \mathbb{E}\left[e^{f(\boldsymbol{\beta})}\right]$ (Eq. 5.24).

7.5 Variational inference of Gaussian components

We optimize $Q(r)$ given $Q(\mu_t)$, then $Q(\mu_t)$ given $Q(r)$. Let the expected sums of residuals,

$$R = \sum_{t=1}^{T} \mathbb{E}\left[\sum_{i\in[n]} z_i \sum_{j\in S_t} (x_{ij}-\mu_t)^2 + s(\mu_t-\mu_0)^2\right].$$

From the general result of mean-field approximation [196], we find

$$\begin{aligned} Q(r) &\propto \exp\left\{\mathbb{E}[\log\mathcal{L} + \log f(\mu|\mathbf{m}_0, s, r)] + \log f(r|c_0, d_0)\right\} \\ &\propto \exp\left\{\log(r)\left(c_0 + T/2 + \sum_{t\in[T]} m_t \mathbb{E}\left[\sum_{i\in[n]} z_i\right]/2 - 1\right) - r(d_0 + R/2)\right\} \\ &= \mathsf{Gam}\left(r\middle|\hat{c},\hat{d}\right). \end{aligned}$$

That are parameterized by

$$\hat{c} \leftarrow c_0 + T/2 + \sum_{t\in[T]} m_t \mathbb{E}\left[\sum_{i\in[n]} z_i\right]/2, \quad \hat{d} \leftarrow d_0 + R/2. \tag{7.2}$$

Again, using the general theory [196] we find $Q(\mu)$ given $Q(r)$.

$$\begin{aligned} Q(\mu) &\propto \exp\left\{\mathbb{E}[\log\mathcal{L} + \log f(\mu|\mu_0, s, r)]\right\} \\ &\propto \exp\left\{-\frac{\mathbb{E}[r]}{2}\sum_{t=1}^{T}\left(m_t\mathbb{E}\left[\sum_{i\in[n]} z_i\right] + s\right)(\mu_t - \hat{\mu}_t)^2\right\} \\ &= \prod_{t=1}^{T} \mathcal{N}\left(\mu_t\middle|\hat{\mu}_t, \gamma_t^{-1}\right) \end{aligned}$$

where

$$\hat{\mu}_t \leftarrow \frac{\mathbb{E}\left[\sum_{i\in[n]} z_i \sum_{j\in S_t} x_{ij}\right] + s\mu_0}{m_t\mathbb{E}\left[\sum_{i\in[n]} z_i\right] + s}$$

and

$$\gamma_t \leftarrow \mathbb{E}[r]\left(m_t\mathbb{E}\left[\sum_{i\in[n]} z_i\right] + s\right).$$

7.6 Locally collapsed variational inference of Gaussian

The goal is straightforward, but tedious. We evaluate

$$\int d\mu_t dr \prod_{i=[n]} \prod_{j \in S_t} \mathcal{N}\left(x_{ij} \middle| \mu_t, r^{-1}\right) \mathcal{N}\left(\mu_t \middle| \hat{\mu}_t, \gamma_t^{-1}\right) \mathsf{Gam}\left(r \middle| \hat{c}, \hat{d}\right).$$

We can factor $\gamma_t = r\hat{s}_t$ with

$$\hat{s}_t \equiv s + \mathbb{E}\left[m_t \sum_{i \in [n]} z_i\right].$$

Let

$$\tilde{m}_t = \frac{\sum_{i=[n]} \sum_{j \in S_t} z_i x_{ti} + \hat{s}_t \hat{\mu}_t}{m_t \sum_{i \in [n]} z_i + \hat{s}_t}.$$

First we integrate out μ.

$$\begin{aligned}
P(\mathbf{x}_i | \hat{\mu}, \hat{s}, r) &= \prod_{t=1}^{T} \int d\mu_t \, \mathcal{N}\left(\mu_t \middle| \hat{\mu}_t, (r\hat{s}_t)^{-1}\right) \prod_{i=1}^{n_t} \mathcal{N}\left(x_{ti} \middle| \mu_t, r^{-1}\right) \\
&= \prod_{t=1}^{T} \int d\mu_t \, (2\pi)^{-\frac{1+n_t}{2}} r^{\frac{1+n_t}{2}} \hat{s}_t^{1/2} \\
&\quad \exp\left\{-\frac{r}{2} \sum_{i=1}^{n_t} (x_{ti} - \mu_t)^2 - \frac{r\hat{s}_t}{2}(\mu_t - \hat{\mu}_t)^2\right\} \\
&= \frac{r^{(T+\sum_t n_t)/2}}{(2\pi)^{(T+\sum_t m_t \sum_{i \in [n]} z_i)/2}} \prod_{t=1}^{T} \hat{s}_t^{1/2} \\
&\quad \exp\left\{-\frac{r}{2}\left[S_t^2 + \hat{s}_t \hat{\mu}_t^2 - \tilde{m}_t^2 \left(m_t \sum_{i \in [n]} z_i + \hat{s}_t\right)\right]\right\} \\
&\quad \int d\mu_t \, \exp\left\{-\frac{r\left(m_t \sum_{i \in [n]} z_i + \hat{s}_t\right)}{2}(\mu_t - \tilde{m}_t)^2\right\} \\
&= \frac{r^{(\sum_t n_t)/2}}{(2\pi)^{(\sum_t m_t \sum_{i \in [n]} z_i)/2}} \\
&\quad \exp\left\{-\frac{r}{2}[\sum_{it} x_{ti}^2 + \sum_t \hat{s}_t \hat{\mu}_t^2 - \sum_t \tilde{m}_t^2 \left(m_t \sum_{i \in [n]} z_i + \hat{s}_t\right)]\right\} \\
&\quad \prod_{t=1}^{T} \sqrt{\frac{\hat{s}_t}{m_t \sum_{i \in [n]} z_i + \hat{s}_t}}.
\end{aligned}$$

Let

$$D \equiv \sum_{it} x_{ti}^2 + \sum_t \hat{s}_t \hat{\mu}_t^2 - \sum_t \tilde{m}_t^2 \left(m_t \sum_{i \in [n]} z_i + \hat{s}_t\right)$$

and

$$n_{tot} \equiv \sum_t m_t \mathbb{E}\left[\sum_{i\in[n]} z_i\right]$$

for simplicity. Finally integrate out r:

$$\begin{aligned} P(\mathbf{x}_i|\hat{\mu},s,\hat{c},\hat{d}) &= \int dr\, P(r|\hat{c},\hat{d})P(\{x_{ti}\}|r,\hat{\mu},\hat{s}) \\ &= \prod_{t=1}^{T}\sqrt{\frac{\hat{s}_t}{m_t\sum_{i\in[n]} z_i+\hat{s}_t}} \\ &\quad (2\pi)^{-n_{tot}/2}\frac{\hat{d}^{\hat{c}}}{\Gamma(\hat{c})}\int dr\, r^{n_{tot}/2}e^{-r\cdot D/2}\, r^{\hat{c}-1}e^{-\hat{d}r} \\ &= \prod_{t=1}^{T}\sqrt{\frac{\hat{s}_t}{m_t\sum_{i\in[n]} z_i+\hat{s}_t}} \\ &\quad (2\pi)^{-n_{tot}/2}\frac{\hat{d}^{\hat{c}}}{\Gamma(\hat{c})}\frac{\Gamma(n_{tot}/2+\hat{c})}{(\hat{d}+D/2)^{n_{tot}/2+\hat{c}}}. \end{aligned}$$

Bibliography

[1] Sumeet Agarwal, Charlotte M Deane, Mason A Porter, and Nick S Jones. Revisiting Date and Party Hubs: Novel Approaches to Role Assignment in Protein Interaction Networks. *PLoS computational biology*, 6(6):e1000817, 2010.

[2] Yong-Yeol Ahn, James P Bagrow, and Sune Lehmann. Link communities reveal multiscale complexity in networks. *Nature*, 466(7307):761–4, August 2010.

[3] Edoardo Airoldi, David Blei, Stephen Fienberg, and Eric Xing. Mixed Membership Stochastic Blockmodels. *The Journal of Machine Learning Research*, 9, June 2008.

[4] Simon Anders and Wolfgang Huber. Differential expression analysis for sequence count data. *Genome biology*, 11(10):R106, 2010.

[5] Charles E Antoniak. Mixtures of Dirichlet Processes with Applications to Bayesian Nonparametric Problems. *The Annals of Statistics*, 2(6):1152–1174, November 1974.

[6] Michelle N Arbeitman, Eileen E M Furlong, Farhad Imam, Eric Johnson, Brian H Null, Bruce S Baker, Mark A Krasnow, Matthew P Scott, Ronald W Davis, and Kevin P White. Gene Expression During the Life Cycle of Drosophila melanogaster. *Science*, 297(5590):2270–2275, 2002.

[7] David A. Bader and Kamesh Madduri. SNAP, Small-world Network Analysis and Partitioning: An open-source parallel graph framework for the exploration of large-scale net-

works. In *2008 IEEE International Symposium on Parallel and Distributed Processing*, pages 1–12. IEEE, April 2008.

[8] Gary D Bader and Christopher W V Hogue. An automated method for finding molecular complexes in large protein interaction networks. *BMC bioinformatics*, 4:2, 2003.

[9] Joel S Bader, Amitabha Chaudhuri, Jonathan M Rothberg, and John Chant. Gaining confidence in high-throughput protein interaction networks. *Nature biotechnology*, 22(1):78–85, 2003.

[10] Brian Ball, Brian Karrer, and M E J Newman. Efficient and principled method for detecting communities in networks. *Physical Review E*, 84(3-2):36103, 2011.

[11] Brian Ball and M. E. J. Newman. Friendship networks and social status. page 7, May 2012.

[12] S Bandyopadhyay, M Mehta, D Kuo, M K Sung, R Chuang, E J Jaehnig, B Bodenmiller, K Licon, W Copeland, M Shales, D Fiedler, J Dutkowski, A Guenole, H van Attikum, K M Shokat, R D Kolodner, W K Huh, R Aebersold, M C Keogh, N J Krogan, and T Ideker. Rewiring of Genetic Networks in Response to DNA Damage. *Science*, 330(6009):1385–1389, 2010.

[13] Eric Banks, Elena Nabieva, Bernard Chazelle, and Mona Singh. Organization of physical interactomes as uncovered by network schemas. *PLoS computational biology*, 4(10):e1000203, October 2008.

[14] Albert-László Barabási and Réka Albert. Emergence of Scaling in Random Networks. *Science*, 286(5439):509–512, 1999.

[15] Artem Barski, Suresh Cuddapah, Kairong Cui, Tae-Young Roh, Dustin E. Schones, Zhibin Wang, Gang Wei, Iouri Chepelev, and Keji Zhao. High-Resolution Profiling of Histone Methylations in the Human Genome. *Cell*, 129(4):823–837, 2007.

[16] Nizar N Batada, Teresa Reguly, Ashton Breitkreutz, Lorrie Boucher, Bobby-Joe Breitkreutz, Laurence D Hurst, and Mike Tyers. Stratus not altocumulus: a new view of the yeast protein interaction network. *PLoS biology*, 4(10):e317, 2006.

[17] Nizar N Batada, Teresa Reguly, Ashton Breitkreutz, Lorrie Boucher, Bobby-Joe Breitkreutz, Laurence D Hurst, and Mike Tyers. Still stratus not altocumulus: further evidence against the date/party hub distinction. *PLoS biology*, 5(6):e154, 2007.

[18] Mohsen Bayati, D Shah, and M Sharma. Max-Product for Maximum Weight Matching: Convergence, Correctness, and LP Duality. *Information Theory, IEEE Transactions on*, 54(3):1241–1251, 2008.

[19] Jonathan W. Berry, Bruce Hendrickson, Randall A. LaViolette, and Cynthia A. Phillips. Tolerating the community detection resolution limit with edge weighting. *Physical Review E*, 83(5):056119, May 2011.

[20] Kenneth Birnbaum, Dennis E Shasha, Jean Y Wang, Jee W Jung, Georgina M Lambert, David W Galbraith, and Philip N Benfey. A gene expression map of the Arabidopsis root. *Science*, 302(5652):1956–60, December 2003.

[21] Yuval Blat and Nancy Kleckner. Cohesins Bind to Preferential Sites along Yeast Chromosome III, with Differential Regulation along Arms versus the Centric Region. *Cell*, 98(2):249–259, 1999.

[22] David M Blei and John D Lafferty. Dynamic topic models. In *ICML '06: Proceedings of the 23rd international conference on Machine learning*. ACM, 2006.

[23] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research (JMLR)*, 3:993–1022, 2003.

[24] U Brandes, D Delling, M Gaertler, R Gorke, M Hoefer, Z Nikoloski, and D Wagner. On Modularity Clustering. *IEEE Transactions on Knowledge and Data Engineering*, 20(2):172–188, 2008.

[25] Ronald L Breiger, Scott A Boorman, and Phipps Arabie. An algorithm for clustering relational data with applications to social network analysis and comparison with multidimensional scaling. *Journal of Mathematical Psychology*, 12(3):328–383, 1975.

[26] Andrew Carlson, Justin Betteridge, Bryan Kisiel, Burr Settles, Estevam R Hruschka, and Tom M Mitchell. Toward an Architecture for Never-Ending Language Learning. *AAAI Conference on Artificial Intelligence; Twenty-Fourth AAAI Conference on Artificial Intelligence*, 2010.

[27] Xiao Chang, Tao Xu, Yun Li, and Kai Wang. Dynamic modular architecture of protein-protein interaction networks beyond the dichotomy of 'date' and 'party' hubs. *Scientific reports*, 3:1691, January 2013.

[28] Myung Soo Cho, Young-Eun Lee, Ji Young Kim, Seungsoo Chung, Yoon Hee Cho, Dae-Sung Kim, Sang-Moon Kang, Haksup Lee, Myung-Hwa Kim, Jeong-Hoon Kim, Joong Woo Leem, Sun Kyung Oh, Young Min Choi, Dong-Youn Hwang, Jin Woo Chang, and Dong-Wook Kim. Highly efficient and large-scale generation of functional dopamine neurons from human embryonic stem cells. *Proceedings of the National Academy of Sciences of the United States of America*, 105(9):3392–7, March 2008.

[29] Aaron Clauset, Cristopher Moore, and M E J Newman. Hierarchical structure and the prediction of missing links in networks. *Nature*, 453(7191):98–101, 2008.

[30] Aaron Clauset, M E J Newman, and Cristopher Moore. Finding community structure in very large networks. *Physical review E, Statistical, nonlinear, and soft matter physics*, 70(6 Pt 2):66111, 2004.

[31] Michael Costanzo, Anastasia Baryshnikova, Jeremy Bellay, Yungil Kim, Eric D Spear, Carolyn S Sevier, Huiming Ding, Judice L Y Koh, Kiana Toufighi, Sara Mostafavi, Jeany Prinz, Robert P St Onge, Benjamin VanderSluis, Taras Makhnevych, Franco J Vizeacoumar, Solmaz Alizadeh, Sondra Bahr, Renee L Brost, Yiqun Chen, Murat Cokol, Raamesh Deshpande, Zhijian Li, Zhen-Yuan Lin, Wendy Liang, Michaela Marback, Jadine Paw, Bryan-Joseph San Luis, Ermira Shuteriqi, Amy Hin Yan Tong, Nydia van Dyk, Iain M Wallace, Joseph A Whitney, Matthew T Weirauch, Guoqing Zhong, Hongwei Zhu, Walid A Houry, Michael Brudno, Sasan Ragibizadeh, Balázs Papp, Csaba Pál, Frederick P Roth, Guri Giaever, Corey Nislow, Olga G Troyanskaya, Howard Bussey, Gary D Bader, Anne-Claude Gingras, Quaid D Morris, Philip M Kim, Chris A Kaiser, Chad L Myers, Brenda J Andrews, and Charles Boone. The genetic landscape of a cell. *Science*, 327(5964):425–431, 2010.

[32] David Croft, Gavin O'Kelly, Guanming Wu, Robin Haw, Marc Gillespie, Lisa Matthews, Michael Caudy, Phani Garapati, Gopal Gopinath, Bijay Jassal, Steven Jupe, Irina Kalatskaya, Shahana Mahajan, Bruce May, Nelson Ndegwa, Esther Schmidt, Veronica Shamovsky, Christina Yung, Ewan Birney, Henning Hermjakob, Peter D'Eustachio, and Lincoln Stein. Reactome: a database of reactions, pathways and biological processes. *Nucleic Acids Res*, 39(Database issue):D691–7, 2011.

[33] Ana Maria Cuervo. The plasma membrane brings autophagosomes to life. *Nature cell biology*, 12(8):735–7, August 2010.

[34] A P Dempster, N M Laird, and D B Rubin. Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, January 1977.

[35] Minghua Deng, Kui Zhang, Shipra Mehta, Ting Chen, and Fengzhu Sun. Prediction of protein function using protein-protein interaction data. *Journal of Computational Biology*, 10(6):947—-960, January 2003.

[36] Marcus T Dittrich, Gunnar W Klau, Andreas Rosenwald, Thomas Dandekar, and Tobias Müller. Identifying functional modules in protein-protein interaction networks: an integrated exact approach. *Bioinformatics*, 24(13):i223–31, 2008.

[37] Janusz Dutkowski, Michael Kramer, Michal A Surma, Rama Balakrishnan, J Michael Cherry, Nevan J Krogan, and Trey Ideker. A gene ontology inferred from molecular networks. *Nature biotechnology*, 31(1):38–45, January 2013.

[38] Jack Edmonds and Richard M. Karp. Theoretical Improvements in Algorithmic Efficiency for Network Flow Problems. *Journal of the ACM*, 19(2):248–264, April 1972.

[39] Bradley Efron. Size, Power and False Discovery Rates. *The Annals of Statistics*, 35(4):1351–1377, August 2007.

[40] Bradley Efron and Robert Tibshirani. On testing the significance of sets of genes. *The Annals of Applied Statistics*, 1(1):107–129, June 2007.

[41] Paul Erdös and Alfréd Rényi. On random graphs I. *Publicationes Mathematicae Debrecen*, 6:290–297, 1959.

[42] Paul Erdös and Alfréd Rényi. On the evolution of random graphs. *Bulletin of the International Statistical Institute*, 38(4):343—-347, 1961.

[43] S Fields and O Song. A novel genetic system to detect protein-protein interactions. *Nature*, 340(6230):245–6, July 1989.

[44] L R Ford and D R Fulkerson. Maximal flow through a network. *Canadian Journal of Mathematics*, 8:399–404, 1956.

[45] Santo Fortunato and Marc Barthelemy. Resolution limit in community detection. *Proc Natl Acad Sci USA*, 104(1):36–41, 2007.

[46] Charless Fowlkes, Serge Belongie, Fan Chung, and Jitendra Malik. Spectral grouping using the Nyström method. *IEEE transactions on pattern analysis and machine intelligence*, 26(2):214–25, February 2004.

[47] Linton C Freeman. A Set of Measures of Centrality Based on Betweenness. *Sociometry*, 40(1):35–41, March 1977.

[48] Jerome Friedman, Trevor Hastie, Holger Höfling, and Robert Tibshirani. Pathwise coordinate optimization. *The Annals of Applied Statistics*, 1(2):302–332, December 2007.

[49] Jerome H Friedman, Trevor Hastie, and Rob Tibshirani. Regularization Paths for Generalized Linear Models via Coordinate Descent. *Journal of Statistical Software*, 33(1):1–22, 2010.

[50] Wenjie Fu, Le Song, and Eric P Xing. Dynamic mixed membership blockmodel for evolving networks. In *ICML '09: Proceedings of the 26th Annual International Conference on Machine Learning*. ACM, 2009.

[51] Anne-Claude Gavin, Markus Bösche, Roland Krause, Paola Grandi, Martina Marzioch, Andreas Bauer, Jörg Schultz, Jens M Rick, Anne-Marie Michon, Cristina-Maria Cruciat, Marita Remor, Christian Höfert, Malgorzata Schelder, Miro Brajenovic, Heinz Ruffner, Alejandro Merino, Karin Klein, Manuela Hudak, David Dickson, Tatjana Rudi, Volker Gnau, Angela Bauch, Sonja Bastuck, Bettina Huhse, Christina Leutwein, Marie-Anne Heurtier, Richard R Copley, Angela Edelmann, Erich Querfurth, Vladimir Rybin, Gerard Drewes, Manfred Raida, Tewis Bouwmeester, Peer Bork, Bertrand Seraphin, Bernhard Kuster, Gitte Neubauer, and Giulio Superti-Furga. Functional organization of the yeast proteome by systematic analysis of protein complexes. *Nature*, 415(6868):141–7, January 2002.

[52] Jane Geisler-Lee, Nicholas O'Toole, Ron Ammar, Nicholas J Provart, A Harvey Millar, and Matt Geisler. A predicted interactome for Arabidopsis. *Plant physiology*, 145(2):317–29, October 2007.

[53] D S Gilmour and J T Lis. In vivo interactions of RNA polymerase II with genes of Drosophila melanogaster. *Mol. Cell. Biol.*, 5(8):2009–2018, August 1985.

[54] L Giot, J S Bader, C Brouwer, A Chaudhuri, B Kuang, Y Li, Y L Hao, C E Ooi, B Godwin, E Vitols, G Vijayadamodar, P Pochart, H Machineni, M Welsh, Y Kong, B Zerhusen, R Malcolm, Z Varrone, A Collis, M Minto, S Burgess, L McDaniel, E Stimpson, F Spriggs, J Williams, K Neurath, N Ioime, M Agee, E Voss, K Furtak, R Renzulli, N Aanensen, S Carrolla, E Bickelhaupt, Y Lazovatsky, A DaSilva, J Zhong, C A Stanyon, R L Finley, K P White, M Braverman, T Jarvie, S Gold, M Leach, J Knight, R A Shimkets, M P McKenna, J Chant, and J M Rothberg. A protein interaction map of Drosophila melanogaster. *Science*, 302(5651):1727–36, December 2003.

[55] M Girvan and M E J Newman. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, 99(12):7821–7826, 2002.

[56] Michel X. Goemans and David P. Williamson. .879-approximation algorithms for MAX CUT and MAX 2SAT. In *Proceedings of the twenty-sixth annual ACM symposium on Theory of computing - STOC '94*, pages 422–431, New York, New York, USA, May 1994. ACM Press.

[57] Michel X. Goemans and David P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM*, 42(6):1115–1145, November 1995.

[58] Debra S Goldberg and Frederick P Roth. Assessing experimentally derived interactions in a small world. *Proceedings of the National Academy of Sciences of the United States of America*, 100(8):4372–6, April 2003.

[59] Benjamin H. Good, Yves-Alexandre de Montjoye, and Aaron Clauset. Performance of modularity maximization in practical contexts. *Physical Review E*, 81(4):046106, April 2010.

[60] Prem Gopalan, David Mimno, Sean M Gerrish, Michael J Freedman, and David M Blei. Scalable Inference of Overlapping Communities. *Advances in Neural Information Procesing Systems (NIPS)*, 2012.

[61] Prem K Gopalan and David M Blei. Efficient discovery of overlapping communities in massive networks. *Proceedings of the National Academy of Sciences of the United States of America*, 110(36):14534–9, September 2013.

[62] David Grünwald, Robert H Singer, and Michael Rout. Nuclear export dynamics of RNA-protein complexes. *Nature*, 475(7356):333–341, 2011.

[63] Aude Guénolé, Rohith Srivas, Kees Vreeken, Ze Zhong Wang, Shuyi Wang, Nevan J. Krogan, Trey Ideker, and Haico van Attikum. Dissection of DNA Damage Responses Using Multiconditional Genetic Interaction Maps. *Molecular Cell*, 49(2):346–358, January 2013.

[64] Jing-Dong J Han, Nicolas Bertin, Tong Hao, Debra S Goldberg, Gabriel F Berriz, Lan V Zhang, Denis Dupuy, Albertha J M Walhout, Michael E Cusick, Frederick P Roth, and Marc Vidal. Evidence for dynamically organized modularity in the yeast protein-protein interaction network. *Nature*, 430(6995):88–93, 2004.

[65] Steve Hanneke, Wenjie Fu, and Eric Xing. Discrete Temporal Models of Social Networks. *Electronic Journal of Statistics*, 4:585–605, 2010.

[66] Leland H. Hartwell, John J. Hopfield, Stanislas Leibler, and Andrew W. Murray. From molecular to modular cell biology. *Nature*, 402, 1999.

[67] W K Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109, April 1970.

[68] Congcong He and Daniel J Klionsky. Regulation mechanisms and signaling pathways of autophagy. *Annual review of genetics*, 43:67–93, January 2009.

[69] Andreas Heider and Rüdiger Alt. virtualArray: a R/bioconductor package to merge raw data from different microarray platforms. *BMC bioinformatics*, 14(1):75, January 2013.

[70] Katherine A Heller and Zoubin Ghahramani. Bayesian Hierarchical Clustering. In *Proceedings of the 22nd international conference on Machine learning - ICML '05*, pages 297–304, New York, New York, USA, 2005. ACM Press.

[71] Keith Henderson, Tina Eliassi-Rad, Spiros Papadimitriou, and Christos Faloutsos. HCDF: A Hybrid Community Discovery Framework. *SDM*, pages 754–765, 2010.

[72] Qirong Ho, Junming Yin, and Eric Xing. On Triangular versus Edge Representations — Towards Scalable Modeling of Networks. In P Bartlett, F C N Pereira, C J C Burges, L Bottou, and K Q Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 2141–2149, 2012.

[73] Matthew D Hoffman, David M Blei, Chong Wang, and John Paisley. Stochastic Variational Inference. *Journal of Machine Learning Research*, 14:1303–1347, 2013.

[74] Jake M Hofman and Chris H Wiggins. Bayesian approach to network modularity. *Physical Review Letters*, 100(25):258701, 2008.

[75] Paul W. Holland, Kathryn Blackmond Laskey, and Samuel Leinhardt. Stochastic blockmodels: First steps. *Social Networks*, 5(2):109–137, 1983.

[76] Paul W Holland and Samuel Leinhardt. Transitivity in Structural Models of Small Groups. *Small Group Research*, 2(2):107–124, May 1971.

[77] Sture Holm. A Simple Sequentially Rejective Multiple Test Procedure. *Scandinavian Journal of Statistics*, 6(2):65–70, January 1979.

[78] Hailiang Huang, Bruno M Jedynak, and Joel S Bader. Where have all the interactions gone? Estimating the coverage of two-hybrid protein interaction maps. *PLoS computational biology*, 3(11):e214, November 2007.

[79] Rafael A Irizarry, Chi Wang, Yun Zhou, and Terence P Speed. Gene set enrichment analysis made simple. *Statistical Methods in Medical Research*, 18(6):565–575, 2009.

[80] T. Ito, K. Tashiro, S. Muta, R. Ozawa, T. Chiba, M. Nishizawa, K. Yamamoto, S. Kuhara, and Y. Sakaki. A comprehensive system to examine two-hybrid interactions in all possible combinations between the yeast proteins. *Proceedings of the National Academy of Sciences*, 97(3):1143–1147, February 2000.

[81] Ronald Jansen, Dov Greenbaum, and Mark Gerstein. Relating Whole-Genome Expression Data with Protein-Protein Interactions. *Genome research*, 12(1):37–46, 2002.

[82] H Jeong, B Tombor, R Albert, Z N Oltvai, and A L Barabási. The large-scale organization of metabolic networks. *Nature*, 407(6804):651–4, October 2000.

[83] David S Johnson, Ali Mortazavi, Richard M Myers, and Barbara Wold. Genome-wide mapping of in vivo protein-DNA interactions. *Science*, 316(5830):1497–502, June 2007.

[84] MichaelI. Jordan, Zoubin Ghahramani, TommiS. Jaakkola, and LawrenceK. Saul. An Introduction to Variational Methods for Graphical Models. *Machine Learning*, 37(2):183–233, 1999.

[85] Tae Hyuk Kang, Ghiabe-Henri Guibinga, H A Jinnah, and Theodore Friedmann. HPRT deficiency coordinately dysregulates canonical Wnt and presenilin-1 signaling: a neurodevelopmental regulatory role for a housekeeping gene? *PloS one*, 6(1):e16572, 2011.

[86] Tae Hyuk Kang, Yongjin Park, Joel S Bader, and Theodore Friedmann. The Housekeeping Gene Hypoxanthine Guanine Phosphoribosyltransferase (HPRT) Regulates Multiple

Developmental and Metabolic Pathways of Murine Embryonic Stem Cell Neuronal Differentiation. *PloS one*, 8(10):e74967, 2013.

[87] David R. Karger and Clifford Stein. A new approach to the minimum cut problem. *Journal of the ACM*, 43(4):601–640, July 1996.

[88] Brian Karrer, Elizaveta Levina, and M E J Newman. Robustness of community structure in networks. *Physical review E, Statistical, nonlinear, and soft matter physics*, 77(4 Pt 2):46119, 2008.

[89] Brian Karrer and M E J Newman. Stochastic blockmodels and community structure in networks. *Physical Review E*, page 11, 2010.

[90] George Karypis and Vipin Kumar. A Fast and High Quality Multilevel Scheme for Partitioning Irregular Graphs. *SIAM J. Sci. Comput.*, 20(1):359–392, 1998.

[91] Robert E Kass and Adrian E Raftery. Bayes Factors. *Journal of the American Statistical Association*, 90(430):773–795, 1995.

[92] Ryan Kelley and Trey Ideker. Systematic interpretation of genetic interactions using protein networks. *Nat Biotech*, 23(5):561–566, 2005.

[93] Myunghwan Kim and Jure Leskovec. Modeling Social Networks with Node Attributes using the Multiplicative Attribute Graph Model. In *Proceedings of the Twenty-Seventh Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-11)*, pages 400–409, Barcelona, Spain, 2011.

[94] Myunghwan Kim and Jure Leskovec. Latent Multi-group Membership Graph Model. In *ICML '12: Proceedings of the 29th international conference on Machine learning*, Stanford University, Stanford, CA 94305, USA, 2012.

[95] Daphne Koller and Nir Friedman. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, 2009.

[96] Kakajan Komurov and Michael White. Revealing static and dynamic modular architecture of the eukaryotic protein interaction network. *Molecular systems biology*, 3:110, 2007.

[97] Risi Imre Kondor and John D. Lafferty. Diffusion Kernels on Graphs and Other Discrete Input Spaces. In *ICML '02 Proceedings of the Nineteenth International Conference on Machine Learning*, pages 315–322. Morgan Kaufmann Publishers Inc., July 2002.

[98] Nevan J Krogan, Gerard Cagney, Haiyuan Yu, Gouqing Zhong, Xinghua Guo, Alexandr Ignatchenko, Joyce Li, Shuye Pu, Nira Datta, Aaron P Tikuisis, Thanuja Punna, Jos\'{e} M Peregr i n Alvarez, Michael Shales, Xin Zhang, Michael Davey, Mark D Robinson, Alberto Paccanaro, James E Bray, Anthony Sheung, Bryan Beattie, Dawn P Richards, Veronica Canadien, Atanas Lalev, Frank Mena, Peter Wong, Andrei Starostine, Myra M Canete, James Vlasblom, Samuel Wu, Chris Orsi, Sean R Collins, Shamanta Chandran, Robin Haw, Jennifer J Rilstone, Kiran Gandi, Natalie J Thompson, Gabe Musso, Peter St Onge, Shaun Ghanny, Mandy H Y Lam, Gareth Butland, Amin M Altaf-Ul, Shigehiko Kanaya, Ali Shilatifard, Erin O'Shea, Jonathan S Weissman, C James Ingles, Timothy R Hughes, John Parkinson, Mark Gerstein, Shoshana J Wodak, Andrew Emili, and Jack F Greenblatt. Global landscape of protein complexes in the yeast Saccharomyces cerevisiae. *Nature*, 440(7084):637–643, 2006.

[99] MK Kuhner and J Felsenstein. A simulation comparison of phylogeny algorithms under equal and unequal evolutionary rates [published erratum appears in Mol Biol Evol 1995 May;12(3):525]. *Mol. Biol. Evol.*, 11(3):459–468, May 1994.

[100] Minjung Kyung, Jeff Gill, Malay Ghosh, and George Casella. Penalized regression, standard errors, and Bayesian lassos. *Bayesian Analysis*, 5(2):369–411, June 2010.

[101] Clara Lago, Elena Clerici, Ludovico Dreni, Christine Horlow, Elisabetta Caporali, Lucia Colombo, and Martin M. Kater. The Arabidopsis TFIID factor AtTAF6 controls pollen tube growth. *Developmental Biology*, 285(1):91–100, 2005.

[102] Andrea Lancichinetti, Santo Fortunato, and J\'{a}nos Kert e sz. Detecting the overlapping and hierarchical community structure in complex networks. *New Journal of Physics*, 11(3):33015, 2009.

[103] Andrea Lancichinetti, Santo Fortunato, and Filippo Radicchi. Benchmark graphs for testing community detection algorithms. *Phys. Rev. E*, 78(4):46110, 2008.

[104] Pascal M Lanctot, Fred H Gage, and Ajit P Varki. The glycans of stem cells. *Current Opinion in Chemical Biology*, 11(4):373–380, August 2007.

[105] Jeffrey T Leek and John D Storey. Capturing heterogeneity in gene expression studies by surrogate variable analysis. *PLoS genetics*, 3(9):1724–35, September 2007.

[106] E L Lehmann and J P Romano. *Testing Statistical Hypotheses*. Springer Texts in Statistics. Springer, 2005.

[107] Jure Leskovec, Jon Kleinberg, and Christos Faloutsos. Graphs over time. *Proceeding of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining - KDD '05*, page 177, 2005.

[108] Jure Leskovec, Kevin J Lang, Anirban Dasgupta, and Michael W Mahoney. Statistical properties of community structure in large social and information networks. In *Proceeding of the 17th international conference on World Wide Web - WWW '08*, page 695, New York, New York, USA, 2008. ACM Press.

[109] Arthur Liberzon, Aravind Subramanian, Reid Pinchback, Helga Thorvaldsdóttir, Pablo Tamayo, and Jill P Mesirov. Molecular signatures database (MSigDB) 3.0. *Bioinformatics*, 27(12):1739–1740, 2011.

[110] Quansheng Liu, Jaclyn C Greimann, and Christopher D Lima. Reconstitution, activities, and structure of the eukaryotic RNA exosome. *Cell*, 127(6):1223–1237, 2006.

[111] John Lonsdale, Jeffrey Thomas, Mike Salvatore, Rebecca Phillips, Edmund Lo, Saboor Shad, Richard Hasz, Gary Walters, Fernando Garcia, Nancy Young, Barbara Foster, Mike Moser, Ellen Karasik, Bryan Gillard, Kimberley Ramsey, Susan Sullivan, Jason Bridge, Harold Magazine, John Syron, Johnelle Fleming, Laura Siminoff, Heather Traino, Maghboeba Mosavel, Laura Barker, Scott Jewell, Dan Rohrer, Dan Maxim, Dana Filkins, Philip Harbach, Eddie Cortadillo, Bree Berghuis, Lisa Turner, Eric Hudson, Kristin Feenstra, Leslie Sobin, James Robb, Phillip Branton, Greg Korzeniewski, Charles Shive, David Tabor, Liqun Qi, Kevin Groch, Sreenath Nampally, Steve Buia, Angela Zimmerman, Anna Smith, Robin Burges, Karna Robinson, Kim Valentino, Deborah Bradbury, Mark Cosentino, Norma Diaz-Mayoral, Mary Kennedy, Theresa Engel, Penelope Williams, Kenyon Erickson, Kristin Ardlie, Wendy Winckler, Gad Getz, David DeLuca, Daniel MacArthur, Manolis Kellis, Alexander Thomson, Taylor Young, Ellen Gelfand, Molly Donovan, Yan Meng, George Grant, Deborah Mash, Yvonne Marcus, Margaret Basile, Jun Liu, Jun Zhu, Zhidong Tu, Nancy J Cox, Dan L Nicolae, Eric R Gamazon, Hae Kyung Im, Anuar Konkashbaev, Jonathan Pritchard, Matthew Stevens, Timothee Flutre, Xiaoquan Wen, Emmanouil T Dermitzakis, Tuuli Lappalainen, Roderic Guigo, Jean Monlong, Michael Sammeth, Daphne Koller, Alexis Battle, Sara Mostafavi, Mark McCarthy, Manual Rivas, Julian Maller, Ivan Rusyn, Andrew Nobel, Fred Wright, Andrey Shabalin, Mike Feolo, Nataliya Sharopova, Anne Sturcke, Justin Paschal, James M Anderson, Elizabeth L Wilder, Leslie K Derr, Eric D Green, Jeffery P Struewing, Gary Temple, Simona Volpi, Joy T Boyer, Elizabeth J Thomson, Mark S Guyer, Cathy Ng, Assya Abdallah, Deborah Colantuoni, Thomas R Insel, Susan E Koester, A Roger Little, Patrick K Bender, Thomas Lehner, Yin Yao, Carolyn C Compton, Jimmie B Vaught, Sherilyn Sawyer, Nicole C Lockhart, Joanne Demchok, and Helen F Moore. The Genotype-Tissue Expression (GTEx) project. *Nature genetics*, 45(6):580–5, July

2013.

[112] Weijun Luo, Michael S Friedman, Kurt D Hankenson, and Peter J Woolf. Time series gene expression profiling and temporal regulatory pathway analysis of BMP6 induced osteoblast differentiation and mineralization. *BMC systems biology*, 5(1):82, January 2011.

[113] Weijun Luo, Michael S Friedman, Kerby Shedden, Kurt D Hankenson, and Peter J Woolf. GAGE: generally applicable gene set enrichment for pathway analysis. *BMC bioinformatics*, 10(1):161, January 2009.

[114] Ulrike Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, December 2007.

[115] Henryk Maciejewski. Gene set analysis methods: statistical models and methodological differences. *Briefings in bioinformatics*, pages bbt002–, February 2013.

[116] J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

[117] Ramamurthy Mani, Robert P St Onge, John L Hartman, Guri Giaever, and Frederick P Roth. Defining genetic interaction. *Proceedings of the National Academy of Sciences of the United States of America*, 105(9):3461–6, March 2008.

[118] Lina Mastrangelo, Ji-Eun Kim, Atsushi Miyanohara, Tae Hyuk Kang, and Theodore Friedmann. Purinergic signaling in human pluripotent stem cells is regulated by the housekeeping gene encoding hypoxanthine guanine phosphoribosyltransferase. *Proc Natl Acad Sci USA*, 109(9):3377–3382, 2012.

[119] Herbert Mayer, Martin Bilban, Vladislav Kurtev, Florian Gruber, Oswald Wagner, Bernd R Binder, and Rainer de Martin. Deciphering regulatory patterns of inflammatory gene expression from interleukin-1-stimulated human endothelial cells. *Arteriosclerosis, thrombosis, and vascular biology*, 24(7):1192–8, July 2004.

[120] G. Mayraz and G.E. Hinton. Recognizing handwritten digits using hierarchical products of experts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(2):189–197, 2002.

[121] Metodi D Metodiev, Nicole Lesko, Chan Bae Park, Yolanda Cámara, Yonghong Shi, Rolf Wibom, Kjell Hultenby, Claes M Gustafsson, and Nils-Göran Larsson. Methylation of 12S rRNA is necessary for in vivo stability of the small subunit of the mammalian mitochondrial ribosome. *Cell metabolism*, 9(4):386–397, 2009.

[122] Tarjei S Mikkelsen, Manching Ku, David B Jaffe, Biju Issac, Erez Lieberman, Georgia Giannoukos, Pablo Alvarez, William Brockman, Tae-Kyung Kim, Richard P Koche, William Lee, Eric Mendenhall, Aisling O'Donovan, Aviva Presser, Carsten Russ, Xiaohui Xie, Alexander Meissner, Marius Wernig, Rudolf Jaenisch, Chad Nusbaum, Eric S Lander, and Bradley E Bernstein. Genome-wide maps of chromatin state in pluripotent and lineage-committed cells. *Nature*, 448(7153):553–60, August 2007.

[123] R Milo, S Shen-Orr, S Itzkovitz, N Kashtan, D Chklovskii, and U Alon. Network Motifs: Simple Building Blocks of Complex Networks. *Science*, 298(5594):824–827, October 2002.

[124] P Mitchell, E Petfalski, A Shevchenko, M Mann, and D Tollervey. The exosome: a conserved eukaryotic RNA processing complex containing multiple 3'–¿5' exoribonucleases. *Cell*, 91(4):457–466, 1997.

[125] Noboru Mizushima, Beth Levine, Ana Maria Cuervo, and Daniel J Klionsky. Autophagy fights disease through cellular self-digestion. *Nature*, 451(7182):1069–75, February 2008.

[126] Sara Mostafavi, Debajyoti Ray, David Warde-Farley, Chris Grouios, and Quaid Morris. GeneMANIA: a real-time multiple association network integration algorithm for predicting gene function. *Genome biology*, 9 Suppl 1:S4, 2008.

[127] Robert F Murphy. An active role for machine learning in drug development. *Nature chemical biology*, 7(6):327–30, June 2011.

[128] Elena Nabieva, Kam Jim, Amit Agarwal, Bernard Chazelle, and Mona Singh. Whole-proteome prediction of protein function via graph-theoretic analysis of interaction maps. *Bioinformatics (Oxford, England)*, 21 Suppl 1(suppl_1):i302–10, June 2005.

[129] Alison V Nairn, Kazuhiro Aoki, Mitche dela Rosa, Mindy Porterfield, Jae-Min Lim, Michael Kulik, J Michael Pierce, Lance Wells, Stephen Dalton, Michael Tiemeyer, and Kelley W Moremen. Regulation of Glycan Structures in Murine Embryonic Stem Cells: COMBINED TRANSCRIPT PROFILING OF GLYCAN-RELATED GENES AND GLYCAN STRUCTURAL ANALYSIS. *The Journal of Biological Chemistry*, 287(45):37835–37856, November 2012.

[130] M. Newman. Clustering and preferential attachment in growing networks. *Physical Review E*, 64(2):025102, July 2001.

[131] M. E. J. Newman. Scientific collaboration networks. II. Shortest paths, weighted networks, and centrality. *Physical Review E*, 64(1):016132, June 2001.

[132] M E J Newman. Modularity and community structure in networks. *Proc Natl Acad Sci USA*, 103(23):8577–8582, June 2006.

[133] M E J Newman and E A Leicht. Mixture models and exploratory analysis in networks. *Proceedings of the National Academy of Sciences of the United States of America*, 104(23):9564–9, June 2007.

[134] Andrew Y Ng, Michael I Jordan, Yair Weiss, and Others. On spectral clustering: Analysis and an algorithm. *Advances in neural information processing systems*, 2:849–856, 2001.

[135] Krzysztof Nowicki and Tom A. B Snijders. Estimation and Prediction for Stochastic Blockstructures. *Journal of the American Statistical Association*, 96(455):1077–1087, September 2001.

[136] Gergely Palla, Imre Derényi, Illés Farkas, and Tamás Vicsek. Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, 435(7043):814–818, 2005.

[137] Konstantina Palla, David Knowles, and Zoubin Ghahramani. An Infinite Latent Attribute Model for Network Data. In John Langford and Joelle Pineau, editors, *Proceedings of the 29th International Conference on Machine Learning (ICML-12)*, ICML '12, pages 1607–1614, New York, 2012. Omnipress.

[138] Xuewen Pan, Ping Ye, Daniel S. Yuan, Xiaoling Wang, Joel S. Bader, and Jef D. Boeke. A DNA Integrity Network in the Yeast Saccharomyces cerevisiae. *Cell*, 124(5):1069–1081, 2006.

[139] Trevor Park and George Casella. The Bayesian Lasso. *Journal of the American Statistical Association*, 103(482):681–686, June 2008.

[140] Yongjin Park and Joel S Bader. Resolving the structure of interactomes with hierarchical agglomerative clustering. *BMC bioinformatics*, 12 Suppl 1:S44, 2011.

[141] Yongjin Park and Joel S Bader. How networks change with time. *Bioinformatics*, 28(12):i40–i48, 2012.

[142] Yongjin Park, Cristopher Moore, and Joel S Bader. Dynamic networks from hierarchical bayesian graph clustering. *PloS one*, 5(1):e8118, 2010.

[143] Yungki Park and Edward M Marcotte. Revisiting the negative example sampling problem for predicting protein-protein interactions. *Bioinformatics (Oxford, England)*, 27(21):3024–8, November 2011.

[144] Yungki Park and Edward M Marcotte. Flaws in evaluation schemes for pair-input computational predictions. *Nature methods*, 9(12):1134–6, December 2012.

[145] Rodolfo M. Pascual, Charlotte K. Billington, Ian P. Hall, Jr. Panettieri, Reynold A., James E. Fish, Stephen P. Peters, and Raymond B. Penn. Mechanisms of cytokine effects on G protein-coupled receptor-mediated signaling in airway smooth muscle. *Am J Physiol Lung Cell Mol Physiol*, 281(6):L1425–1435, December 2001.

[146] Lucy F Pemberton, Günter Blobel, and Jonathan S Rosenblum. Transport routes through the nuclear pore complex. *Current Opinion in Cell Biology*, 10(3):392–399, 1998.

[147] T C Petrossian and S G Clarke. Multiple Motif Scanning to Identify Methyltransferases from the Yeast Proteome. *Molecular & Cellular Proteomics*, 8(7):1516–1526, 2009.

[148] Suzanne R Pfeffer. Unconventional secretion by autophagosome exocytosis. *The Journal of cell biology*, 188(4):451–2, February 2010.

[149] Patrick C Phillips. Epistasis–the essential role of gene interactions in the structure and evolution of genetic systems. *Nature reviews. Genetics*, 9(11):855–67, November 2008.

[150] Shuye Pu, Jessica Wong, Brian Turner, Emerson Cho, and Shoshana J Wodak. Up-to-date catalogues of yeast protein complexes. *Nucleic Acids Res*, 37(3):825–831, 2009.

[151] Yan Qi, Yasir Suhail, Yu-yi Lin, Jef D Boeke, and Joel S Bader. Finding friends and enemies in an enemies-only network: a graph diffusion kernel for predicting novel genetic interactions and co-complex membership from yeast genetic interactions. *Genome research*, 18(12):1991–2004, 2008.

[152] Jian Qiu and William Stafford Noble. Predicting co-complexed protein pairs from heterogeneous data. *PLoS computational biology*, 4(4):e1000054, April 2008.

[153] E Ravasz, A L Somera, D A Mongru, Z N Oltvai, and A L Barabási. Hierarchical organization of modularity in metabolic networks. *Science*, 297(5586):1551–5, August 2002.

[154] Erzsébet Ravasz and Albert-László Barabási. Hierarchical organization in complex networks. *Physical Review E*, 67(2):026112, February 2003.

[155] Brinda Ravikumar, Kevin Moreau, Luca Jahreiss, Claudia Puri, and David C Rubinsztein. Plasma membrane contributes to the formation of pre-autophagosomal structures. *Nature cell biology*, 12(8):747–57, August 2010.

[156] G Rigaut, A Shevchenko, B Rutz, M Wilm, M Mann, and B Séraphin. A generic protein purification method for protein complex characterization and proteome exploration. *Nature biotechnology*, 17(10):1030–2, October 1999.

[157] Alexander W Rives and Timothy Galitski. Modular organization of cellular networks. *Proceedings of the National Academy of Sciences of the United States of America*, 100(3):1128–33, February 2003.

[158] Assen Roguev, Dale Talbot, Gian Luca Negri, Michael Shales, Gerard Cagney, Sourav Bandyopadhyay, Barbara Panning, and Nevan J Krogan. Quantitative genetic-interaction mapping in mammalian cells. *Nature methods*, 10(5):432–7, May 2013.

[159] Karl Rohe, Sourav Chatterjee, and Bin Yu. Spectral clustering and the high-dimensional stochastic blockmodel. *The Annals of Statistics*, 39(4):1878–1915, August 2011.

[160] Lo\"{\i}c Royer, Matthias Reimann, Bill Andreopoulos, and Michael Schroeder. Unraveling protein networks with power graph analysis. *PLoS computational biology*, 4(7):e1000108, 2008.

[161] Jean-François Rual, Kavitha Venkatesan, Tong Hao, Tomoko Hirozane-Kishikawa, Amélie Dricot, Ning Li, Gabriel F Berriz, Francis D Gibbons, Matija Dreze, Nono Ayivi-Guedehoussou, Niels Klitgord, Christophe Simon, Mike Boxem, Stuart Milstein, Jennifer Rosenberg, Debra S Goldberg, Lan V Zhang, Sharyl L Wong, Giovanni Franklin, Siming Li, Joanna S Albala, Janghoo Lim, Carlene Fraughton, Estelle Llamosas, Sebiha Cevik,

Camille Bex, Philippe Lamesch, Robert S Sikorski, Jean Vandenhaute, Huda Y Zoghbi, Alex Smolyar, Stephanie Bosak, Reynaldo Sequerra, Lynn Doucette-Stamm, Michael E Cusick, David E Hill, Frederick P Roth, and Marc Vidal. Towards a proteome-scale map of the human protein-protein interaction network. *Nature*, 437(7062):1173–8, October 2005.

[162] Colm J. Ryan, Assen Roguev, Kristin Patrick, Jiewei Xu, Harlizawati Jahari, Zongtian Tong, Pedro Beltrao, Michael Shales, Hong Qu, Sean R. Collins, Joseph I. Kliegman, Lingli Jiang, Dwight Kuo, Elena Tosti, Hyun-Soo Kim, Winfried Edelmann, Michael-Christopher Keogh, Derek Greene, Chao Tang, Pádraig Cunningham, Kevan M. Shokat, Gerard Cagney, J. Peter Svensson, Christine Guthrie, Peter J. Espenshade, Trey Ideker, and Nevan J. Krogan. Hierarchical Modularity and the Evolution of Genetic Interactomes across Species. *Molecular Cell*, 46(5):691–704, June 2012.

[163] GB Rybicki and DG Hummer. An accelerated lambda iteration method for multilevel radiative transfer. I - Non-overlapping lines with background continuum. *Astronomy and Astrophysics*, 245(1):171–181, 1991.

[164] C Saveanu. Identification of 12 New Yeast Mitochondrial Ribosomal Proteins Including 6 That Have No Prokaryotic Homologues. *Journal of Biological Chemistry*, 276(19):15861–15867, 2001.

[165] G Schlenstedt, E Smirnova, R Deane, J Solsbacher, U Kutay, D Görlich, H Ponstingl, and F R Bischoff. Yrb4p, a yeast ran-GTP-binding protein involved in import of ribosomal protein L25 into the nucleus. *The EMBO journal*, 16(20):6237–6249, 1997.

[166] M Schuldiner, S R Collins, J S Weissman, and N J Krogan. Quantitative genetic analysis in Saccharomyces cerevisiae using epistatic miniarray profiles (E-MAPs) and its application to chromatin functions. *Methods*, 40(4):344–352, December 2006.

[167] Gideon Schwarz. Estimating the Dimension of a Model. *The Annals of Statistics*, 6(2):461–464, March 1978.

[168] Bernhard Schweighofer, Julia Testori, Caterina Sturtzel, Susanne Sattler, Herbert Mayer, Oswald Wagner, Martin Bilban, and Erhard Hofer. The VEGF-induced transcriptional response comprises gene clusters at the crossroad of angiogenesis and inflammation. *Thrombosis and haemostasis*, 102(3):544–54, September 2009.

[169] Roded Sharan, Igor Ulitsky, and Ron Shamir. Network-based prediction of protein function. *Molecular systems biology*, 3:88, January 2007.

[170] Tal Shay, Vladimir Jojic, Or Zuk, Katherine Rothamel, David Puyraimond-Zemmour, Ting Feng, Ei Wakamatsu, Christophe Benoist, Daphne Koller, Aviv Regev, and ImmGen Consortium. Conservation and divergence in the transcriptional programs of the human and mouse immune systems. *Proc Natl Acad Sci USA*, 110(8):2946–2951, 2013.

[171] Tom A.B. Snijders and Krzysztof Nowicki. Estimation and Prediction for Stochastic Blockmodels for Graphs with Latent Block Structure. *Journal of Classification*, 14(1):75–100, January 1997.

[172] Le Song, Mladen Kolar, and Eric P Xing. KELLER: estimating time-varying interactions between genes. *Bioinformatics*, 25(12):i128–36, 2009.

[173] Victor Spirin and Leonid A Mirny. Protein complexes and functional modules in molecular networks. *Proceedings of the National Academy of Sciences of the United States of America*, 100(21):12123–8, October 2003.

[174] Chris Stark, Bobby-Joe Breitkreutz, Teresa Reguly, Lorrie Boucher, Ashton Breitkreutz, and Mike Tyers. BioGRID: a general repository for interaction datasets. *Nucleic Acids Res*, 34(Database Issue):D535, 2006.

[175] Caterina Strambio-De-Castillia, Mario Niepel, and Michael P Rout. The nuclear pore complex: bridging nuclear transport and gene regulation. *Nature reviews. Molecular cell biology*, 11(7):490–501, 2010.

[176] Aravind Subramanian, Pablo Tamayo, Vamsi K Mootha, Sayan Mukherjee, Benjamin L Ebert, Michael A Gillette, Amanda Paulovich, Scott L Pomeroy, Todd R Golub, Eric S Lander, and Jill P Mesirov. Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. *Proceedings of the National Academy of Sciences of the United States of America*, 102(43):15545–50, October 2005.

[177] W Sun and T Tony Cai. Largescale multiple testing under dependence. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 71(2):393–424, 2009.

[178] Wenguang Sun and T Tony Cai. Oracle and Adaptive Compound Decision Rules for False Discovery Rate Control. *Journal of the American Statistical Association*, 102(479):901–912, 2007.

[179] Yosuke Tamada, Kazuki Nakamori, Hiromi Nakatani, Kentaro Matsuda, Shingo Hata, Tsuyoshi Furumoto, and Katsura Izui. Temporary expression of the TAF10 gene and its requirement for normal development of Arabidopsis thaliana. *Plant & cell physiology*, 48(1):134–46, January 2007.

[180] Ian W Taylor, Rune Linding, David Warde-Farley, Yongmei Liu, Catia Pesquita, Daniel Faria, Shelley Bull, Tony Pawson, Quaid Morris, and Jeffrey L Wrana. Dynamic modularity in protein interaction networks predicts breast cancer outcome. *Nat Biotechnol*, 27(2):199–204, 2009.

[181] Robert Tibshirani and Pei Wang. Spatial smoothing and hot spot detection for CGH data using the fused lasso. *Biostatistics (Oxford, England)*, 9(1):18–29, January 2008.

[182] Amy Hin Yan Tong, Marie Evangelista, Ainslie B Parsons, Hong Xu, Gary D Bader, Nicholas Pagé, Mark Robinson, Sasan Raghibizadeh, Christopher W V Hogue, Howard Bussey, Brenda Andrews, Mike Tyers, and Charles Boone. Systematic Genetic Analysis with Ordered Arrays of Yeast Deletion Mutants. *Science*, 294(5550):2364–2368, December 2001.

[183] Amy Hin Yan Tong, Guillaume Lesage, Gary D Bader, Huiming Ding, Hong Xu, Xiaofeng Xin, James Young, Gabriel F Berriz, Renee L Brost, Michael Chang, YiQun Chen, Xin Cheng, Gordon Chua, Helena Friesen, Debra S Goldberg, Jennifer Haynes, Christine Humphries, Grace He, Shamiza Hussein, Lizhu Ke, Nevan Krogan, Zhijian Li, Joshua N Levinson, Hong Lu, Patrice Ménard, Christella Munyana, Ainslie B Parsons, Owen Ryan, Raffi Tonikian, Tania Roberts, Anne-Marie Sdicu, Jesse Shapiro, Bilal Sheikh, Bernhard Suter, Sharyl L Wong, Lan V Zhang, Hongwei Zhu, Christopher G Burd, Sean Munro, Chris Sander, Jasper Rine, Jack Greenblatt, Matthias Peter, Anthony Bretscher, Graham Bell, Frederick P Roth, Grant W Brown, Brenda Andrews, Howard Bussey, and Charles Boone. Global Mapping of the Yeast Genetic Interaction Network. *Science*, 303(5659):808–813, February 2004.

[184] K J Tracey and A Cerami. Tumor necrosis factor, other cytokines and disease. *Annual review of cell biology*, 9:317–43, January 1993.

[185] Cole Trapnell, Lior Pachter, and Steven L Salzberg. TopHat: discovering splice junctions with RNA-Seq. *Bioinformatics*, 25(9):1105–1111, 2009.

[186] Benjamin P Tu, Andrzej Kudlicki, Maga Rowicka, and Steven L McKnight. Logic of the yeast metabolic cycle: temporal compartmentalization of cellular processes. *Science*, 310(5751):1152–1158, 2005.

[187] P Uetz, L Giot, G Cagney, T A Mansfield, R S Judson, J R Knight, D Lockshon, V Narayan, M Srinivasan, P Pochart, A Qureshi-Emili, Y Li, B Godwin, D Conover, T Kalbfleisch, G Vijayadamodar, M Yang, M Johnston, S Fields, and J M Rothberg. A comprehensive analysis of protein-protein interactions in Saccharomyces cerevisiae. *Nature*, 403(6770):623–627, 2000.

[188] Stijn Van Dongen. Graph Clustering Via a Discrete Uncoupling Process. *SIAM Journal on Matrix Analysis and Applications*, 30(1):121–141, January 2008.

[189] Youichiro Wada, Yoshihiro Ohta, Meng Xu, Shuichi Tsutsumi, Takashi Minami, Kenji Inoue, Daisuke Komura, Jun'ichi Kitakami, Nobuhiko Oshida, Argyris Papantonis, Akashi Izumi, Mika Kobayashi, Hiroko Meguro, Yasuharu Kanki, Imari Mimura, Kazuki Yamamoto, Chikage Mataki, Takao Hamakubo, Katsuhiko Shirahige, Hiroyuki Aburatani, Hiroshi Kimura, Tatsuhiko Kodama, Peter R Cook, and Sigeo Ihara. A wave of nascent transcription on activated human genes. *Proceedings of the National Academy of Sciences of the United States of America*, 106(43):18357–61, October 2009.

[190] Chong Wang and David M Blei. Truncation-free Online Variational Inference for Bayesian Nonparametric Models. In P. Bartlett, F.C.N. Pereira, C.J.C. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 422–430, 2012.

[191] Chong Wang and David M. Blei. Variational Inference in Nonconjugate Models. *Journal of Machine Learning Research*, 14:1005–1031, September 2013.

[192] Stanley Wasserman and Carolyn Anderson. Stochastic a posteriori blockmodels: Construction and assessment. *Social Networks*, 9(1):1–36, 1987.

[193] D J Watts and S H Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393(6684):440–442, 1998.

[194] Harrison C White, Scott A Boorman, and Ronald L Breiger. Social Structure from Multiple Networks. I. Blockmodels of Roles and Positions. *American Journal of Sociology*, 81(4):730–780, January 1976.

[195] Zhijin Wu, Rafael A Irizarry, Robert Gentleman, Francisco Martinez-Murillo, and Forrest Spencer. A Model-Based Background Adjustment for Oligonucleotide Expression Arrays. *Journal of the American Statistical Association*, 99(468):909–917, 2004.

[196] Eric P Xing, Michael I Jordan, and Stuart Russell. A generalized mean field algorithm for variational inference in exponential families. In *Uncertainty in Artificial Intelligence (UAI 2003)*, Computer Science Division, Unifersity of California Berkeley, 2003.

[197] Tetsuro Yamamoto and Yasuhiko Ikebe. Inversion of band matrices. *Linear Algebra and its Applications*, 24(0):105–111, April 1979.

[198] Ping Ye, Brian D Peyser, Xuewen Pan, Jef D Boeke, Forrest A Spencer, and Joel S Bader. Gene function prediction from congruent synthetic lethal interactions in yeast. *Molecular systems biology*, 1(1), 2005.

[199] J S Yedidia, W T Freeman, and Y Weiss. Constructing free-energy approximations and generalized belief propagation algorithms. *Information Theory, IEEE Transactions on*, 51(7):2282–2312, 2005.

[200] C J Yoo and S L Wolin. La proteins from Drosophila melanogaster and Saccharomyces cerevisiae: a yeast homolog of the La autoantigen is dispensable for growth. *Molecular and cellular biology*, 14(8):5412–5424, 1994.

[201] Haiyuan Yu, Pascal Braun, Muhammed A Yildirim, Irma Lemmens, Kavitha Venkatesan, Julie Sahalie, Tomoko Hirozane-Kishikawa, Fana Gebreab, Na Li, Nicolas Simonis, Tong Hao, Jean-Francois Rual, Amelie Dricot, Alexei Vazquez, Ryan R Murray, Christophe Simon, Leah Tardivo, Stanley Tam, Nenad Svrzikapa, Changyu Fan, Anne-Sophie de Smet, Adriana Motyl, Michael E Hudson, Juyong Park, Xiaofeng Xin, Michael E Cusick, Troy Moore, Charlie Boone, Michael Snyder, Frederick P Roth, Albert-L\'{a}szl\'{o} Barabasi, Jan Tavernier, David E Hill, and Marc Vidal. High-Quality Binary Protein Interaction Map of the Yeast Interactome Network. *Science*, 322(5898):104–110, 2008.

[202] Wayne W Zachary. An Information Flow Model for Conflict and Fission in Small Groups. *Journal of Anthropological Research*, 33(4):452–473, 1977.

[203] Junjun Zhang, Syed Haider, Joachim Baran, Anthony Cros, Jonathan M Guberman, Jack Hsu, Yong Liang, Long Yao, and Arek Kasprzyk. BioMart: a data federation framework for large collaborative projects. *Database : the journal of biological databases and curation*, 2011(0):bar038, January 2011.

[204] H Zhou, H Ma, W Wei, D Ji, X Song, J Sun, J Zhang, and L Jia. B4GALT family mediates the multidrug resistance of human leukemia cells by regulating the hedgehog pathway and the expression of p-glycoprotein and multidrug resistance-associated protein 1. *Cell death & disease*, 4:e654, January 2013.

Chapter 8

CURRICULUM VITAE

YONGJIN PARK

February 18, 2014

Educational History:

Ph.D.	2014	Biomedical Engineering Mentor: Joel. S. Bader, Ph.D.	Johns Hopkins School of Medicine
M.S.	2008	Computational Biology	Carnegie Mellon University
B.S.	2006	Biological Science	Seoul National University

Publications

Kang T, Park Y, Bader JS, Friedmann T, *The housekeeping gene HPRT regulates multiple developmental pathways during neural differentiation of murine embryonic stem cells*, PLoS One

Park Y and Bader JS, *How networks change with time*, Bioinformatics. 2012 Jan 15;28(12)

Park Y and Bader JS, *Resolving the structure of interactomes with hierarchical agglomerative clustering*, BMC Bioinformatics, vol. 12, p. S44, 2011.

Park Y, Moore C, and Bader JS, *Dynamic networks from hierarchical bayesian graph clustering*, PLoS ONE, vol. 5, no. 1, p. e8118, 2010.

Park Y, Shackney S, and Schwartz R, *Network-based inference of cancer progression from microarray data*, IEEE/ACM Trans Comput Biol Bioinform, vol. 6, no. 2, pp. 200-212, 2009.

Talks ($\star$) / Posters ($\bullet$)

$\star$ *Discriminative gene set enrichment: making sense out of 1000 network modules*, Technology Center for Networks and Pathways, Johns Hopkins School of Medicine, Baltimore, MD, Nov 12, 2013

$\star$ *How networks change with time*, Intelligent Systems for Molecular Biology, 2012, Long Beach, CA, Jul 15, 2012

$\bullet$ *Dynamics of protein complexes in stem cell differentiation*, Technology Center for Networks and Pathways, Bethesda, MD, Jul 8, 2012

$\star$ *Resolving the structure of interactome with hierarchical agglomerative clustering*, Joint Statistical Meetings, Miami Beach, Florida, Jul 30- Aug 4, 2011

$\bullet$ *Protein complex dynamics in the metabolic cycle*, Technology Center for Networks and Pathways all hands meeting, Rockville, MD, Apr 14-15, 2011

$\star$ *Resolving the structure of interactome with hierarchical agglomerative clustering*, Asia Pacific Bioinformatics Conference, Incheon, Korea, Jan 11-14, 2011

$\star$ *Dynamic networks from hierarchical Bayesian graph clusstering*, RECOMB Systems Biology, Boston, MA, Dec 2-6, 2009

$\bullet$ *Network Dynamics in Space and Time*, Microsoft eScience workshop, Pittsburgh, PA, October 15-17, 2009